Putting Humans in a Scene: Learning Affordance in 3D Indoor Environments

Xueting Li*[1], Sifei Liu[2], Kihwan Kim[2], Xiaolong Wang[3], Ming-Hsuan Yang[1], and Jan Kautz[2]

[1]University of California, Merced, [2]NVIDIA, [3]Carnegie Mellon University

Abstract

Affordance[1] modeling plays an important role in visual understanding. In this paper, we aim to predict affordances of 3D indoor scenes, specifically what human poses are afforded by a given indoor environment, such as sitting on a chair or standing on the floor. In order to predict valid affordances and learn possible 3D human poses in indoor scenes, we need to understand the semantic and geometric structure of a scene as well as its potential interactions with a human. To learn such a model, a large-scale dataset of 3D indoor affordances is required. In this work, we build a fully automatic 3D pose synthesizer that fuses semantic knowledge from a large number of 2D poses extracted from TV shows as well as 3D geometric knowledge from voxel representations of indoor scenes. With the data created by the synthesizer, we introduce a 3D pose generative model to predict semantically plausible and physically feasible human poses within a given scene (provided as a single RGB, RGB-D, or depth image). We demonstrate that our human affordance prediction method consistently outperforms existing state-of-the-art methods. The project website can be found at `https://sites.google.com/view/3d-affordance-cvpr19`.

1. Introduction

There is a long history of studies on functional reasoning of objects and scenes. Instead of focusing on the semantics of objects and scenes, Gibson proposes the idea of affordances [5], which can be seen as the "opportunities for interactions" with the environment.

To infer the affordances of objects and scenes, researchers have studied the explicit modeling of physical interactions and contacts between human and the 3D scene through simulations [35, 23, 7]. For example, Zhu et al. [35] explicitly model sitting styles by inferring the forces and pressures from the interaction between humans and objects in a scene. However, explicit modeling suffers from the problem of generalization for other types of poses. To tackle the problem of generalization, researchers have proposed to directly infer affordances in a data-driven manner [4, 3, 27]. Specifically, Wang et al. [27] design a method to collect human-scene interactions by processing video frames of various TV shows and train CNNs for affordance reasoning. Though the method is able to generate semantically plausible human poses aligned with scene images, it is not able to follow the geometry of the 3D world and often pro-

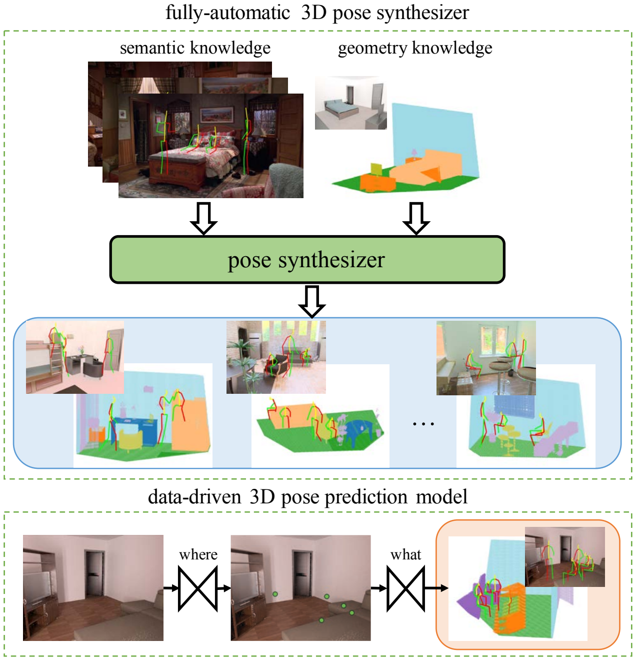

fully-automatic 3D pose synthesizer
semantic knowledge
geometry knowledge
pose synthesizer
...
data-driven 3D pose prediction model
where
what

Figure 1. **Overview of the proposed method.** Our method contains two stages. First, we propose a fully-automatic 3D pose synthesizer, which can synthesize an infinite number of 3D poses for indoor scenes (see Section 3). We illustrate synthesized pose samples in the *light blue box*. Second, we learn an end-to-end 3D affordance prediction model by jointly learning the distribution of locations and 3D poses (see Section 4). We show generated poses in the *light orange box*. Zoom-in to see details.

*This work is completed during an internship at NVIDIA.

[1] Affordances are opportunities for interactions in a scene or environment. It represents what interactions an environment could provide for humans, e.g., a chair provides the opportunity to sit.

duces results violating physics (e.g., first row in Fig. 10) due to a lack of 3D geometric information of the scenes (as the data consists only of video frames and 2D poses).

In this paper, our goal is to learn a model that is able to generate 3D human poses that not only follow natural human behaviors (e.g., humans should sit rather than stand on a chair), but also are physically feasible (e.g., humans should not collide with objects). To achieve this goal, we need to synthesize an appropriate dataset containing human poses in various indoor scenes. We first train a 2D pose prediction model using an existing real-world video dataset [27]. The trained model is then adapted to the indoor images in the SUNCG dataset [26, 30], which contains complete 3D annotations, e.g., camera parameters and 3D geometry (we use a voxel representation). Since there exist well-defined links between the 2D images and the 3D world, given these annotations, we can map the generated 2D poses into the 3D world. We further adjust these mapped poses in 3D voxel space to make sure they are physically feasible (no intersections with objects and well supported by surrounding furniture). Our dataset synthesis approach is fully automatic and can synthesize numerous, diverse "ground-truth" poses in different locations.

Given this large amount of data, we are able to train an affordance prediction model, which aims to generate 3D human poses given a single scene image. We model the pose distributions conditioned on the scene context, where the pose distributions are factorized into the distributions of (a) pose pelvis joint locations, and (b) pose appearance on top of sampled locations. We name them the *where* and *what* modules, respectively. The two modules are jointly trained using the pose pelvis joint locations as a differentiable bridge. Essentially, we propose a geometry-aware discriminator to encourage the model to better understand the geometry of the scene (see Fig. 4 (b)), even through a single RGB image. We evaluate the plausibility of our generated 3D poses via user study as well as a trained classifier that aims to score the "authenticity" of generated poses. We also map generated poses back to the 3D voxel space to evaluate their physical correctness in the 3D world.

Our main contributions can be summarized as:

- We propose an efficient, fully-automatic 3D human pose synthesizer that leverages the pose distributions learned from the 2D world, and the physical feasibility extracted from the 3D world.
- We develop a generative model for 3D affordance prediction which generates plausible human poses with full 3D information, from a single scene image.
- We set a new benchmark for large-scale human-centric affordance prediction on the SUNCG dataset by leveraging the human pose synthesizer and the pose generator.

2. Related Work

Scene understanding. In recent years, much progress has been made [28, 1, 31] in the field of semantic scene understanding thanks to large-scale labeled datasets [33, 18]. A few methods [6, 29, 19] aim to specially model human-scene interactions. However, they focus on detecting human-object interactions rather than explicitly reasoning about object functionality in a scene.

Object functionality reasoning. For deeper reasoning of objects in a scene beyond the conventional scene understanding techniques, several approaches [7, 35, 32, 36] revisit the principle of affordance [5] via explicitly modeling the functionality of objects in a scene. For instance, Grabner et al. [7] propose to detect a chair by considering its functionality (i.e. examining whether an imaginary human can sit on the object). Zhu et al. [36] recognize tools and infer their functionality by analyzing RGB-D videos. However, these methods are hard to generalize to real-world scenarios because they rely heavily on complete 3D geometry information of a scene.

Human affordance prediction. Other than explicitly modeling object functionality, several recent algorithms [15, 2, 34, 14] exploit human affordance in a data-driven manner. Gupta et al. [8] manually associate human actions with exemplar poses and search feasible locations for those actions in a scene by performing 3D correlation between poses and scene voxels. Fouhey et al. [3] propose to estimate human-scene interactions and scene geometry by observing human actions in time-lapse sequences. Roy and Todorovic [24] predict affordance segmentation maps for specific actions from single images by predicting and fusing mid-level visual cues. Wang et al. [27] collect human-scene and human-object interactions by scanning through millions of video frames in different TV series and train CNNs for human affordance reasoning, which partly motivated our work. However, the data collection process still requires manual effort, and can only collect limited training examples ($\sim$20K). Without sufficient data and geometric knowledge of scenes, it is hard for CNNs to follow the geometric constraints of a scene, leading to results that often violate the physics.

Instance placement in a scene. Our affordance prediction method which puts humans into feasible locations in a scene can be seen as an instance placement task. Several recent approaches [17, 21, 16] focus on predicting either location or appearance of an instance in a scene. For example, Lin et al. [17] propose to insert objects into feasible locations in a scene. However, this method requires a user provided template as the instance. Ouyang et al. [21] utilize a Generative Adversarial Network to in-paint pedestrians at given locations in a scene. Closest to our work, Lee [16] jointly model a context-aware distribution of the location and shape of object instances given a scene. Nevertheless, their method focuses on inserting instances in 2D images and does not consider any physical feasibility in 3D scenes.

3. 3D Pose Synthesis

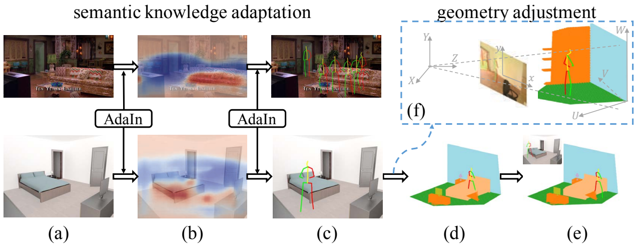


Figure 2. **Pose synthesis.** (a) Input image. (b) Location heat map. The blue and red regions denote the areas suitable for standing and sitting. (c) Generated pose. (d) Corresponding pose in the voxel. (e) Adjusted pose in voxel. (f) Mapping from image to voxel.

Collecting a large-scale dataset of human poses with 3D scene annotations is currently a tedious task [22]. In this section, we show how to automatically synthesize "ground-truth" 3D human poses in various indoor scenes. To ensure the correctness of generated poses, we take two factors into account: (i) semantic plausibility; the synthesized poses should follow natural human behaviours in typical indoor environments, and (ii) physical correctness; the human poses should not collide with objects in a scene or float in the air. To satisfy constraint (i), we learn a 2D human pose generative model that encodes the natural human pose distributions from existing 2D examples [27] (see Fig. 2(a) to (c) and Section 3.1). Then, given the camera parameters, we map the generated poses into the 3D world represented as voxels. (see Fig. 2(f)) and Section 3.2). Finally, we introduce an efficient way to adjust the poses in the 3D scene to satisfy constraint (ii) (see Fig. 2(d) to (e) and Section 3.3).

Overall, we use our pose synthesizer to produce around 1.5 million "ground-truth" poses, which are then used in Section 4. Fig. 1 (light blue box) shows samples of poses obtained by our pose synthesizer in 3D space and their projections onto 2D images.

3.1. Affordance Prediction in 2D Scene Images

We synthesize 3D human poses by first generating poses in 2D images, then projecting them into the 3D world as shown in Fig. 2. To this end, we utilize the Sitcom dataset [27] which contains pose samples captured from sitcom videos and train a human pose prediction model. Then we adapt the trained model onto the SUNCG images to generate poses that follow natural human behaviors. The work by Wang et al. [27] only focuses on predicting the most plausible human pose at a feasible location in 2D scene images. However, the annotations of such feasible locations are not available in the SUNCG dataset. Therefore, we need to learn a network that predicts locations to put humans in a scene, before utilizing the method in [27] to generate human poses at each predicted location.

We represent each pose location by its pelvis joint coordinates. A typical technique [24] for predicting human pose locations is to learn a pixel-wise probability map of a scene. However, the existing 2D pose annotations are highly sparse (typically only a few poses per scene). To address this issue, we augment the annotation from a single point to a local square patch, assuming the nearby area can afford the same pose. Furthermore, Wang et al. [27] cluster all poses into 30 clusters according to their gestures and feed the cluster center corresponding to each pose as a condition to their pose prediction model. Thus to utilize their pose prediction model, we not only need to find feasible locations for human poses, but also predict the most likely pose class at each predicted location.

To this end, for each location that has a pose annotation, we use a 31-dimensional binary vector to represent the corresponding pose class. Locations without pose annotations are labeled as background (the 31^{st} class). This results in a $31 \times h \times w$ pose location map as the ground truch heatmap for each scene, where h and w are the height and width of the scene image. We learn a CNN that takes a scene image as input and predicts the corresponding heat map. During the testing process, we sample from the heat map and output both locations possible for human poses as well as the most likely pose class at these locations.

Since our ultimate goal is to generate 3D poses, we first map 2D pose annotations in the Sitcom dataset to 3D poses in the Human3.6M dataset [11] and then train the pose generation model in [27] to generate 3D poses. Detailed mapping process can be found in the appendix. In this way, we extend the pose prediction in [27] from generating 2D poses in given ground truth locations, to generating 3D poses at sampled locations. Fig. 2(b) and (c) illustrate location heat maps and poses predicted by our model respectively.

To narrow the domain gap between the SUNCG and the Sitcom dataset, we perform domain adaptation [10] when applying the trained model onto the SUNCG images, via matching the second-order statistics of image features for both location and pose prediction models. More details about the domain adaptation can be found in the appendix.

3.2. Mapping Poses into 3D Scenes

Mapping a pixel from the image coordinates to the 3D world requires its depth value and the camera parameters. Unfortunately, depth values are not known for the generated human poses. However, we circumvent this problem by estimating these depth values from the known real-world distribution of human heights. We sample the height of a human for standing pose from $\mathcal{N}(1.65, 0.1)$, and for sitting pose from $\mathcal{N}(1.20, 0.1)$. Given the sampled human height in 3D world, we can estimate the depth d of each pose by $d = \frac{H \times f}{H_p \times r_{32}}$, where H_p is the pose height at pixel coordinate system, H is the sampled human height mentioned

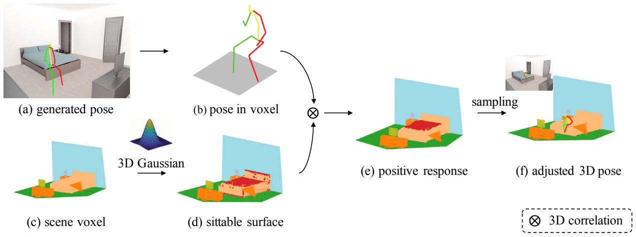


Figure 3. **Affordance adjustment.** (a) Generated pose by the model described in Section 3.1. (b) Corresponding pose in the voxel space. (c) A scene voxel. (d) The surface of a bed (colored in red) detected by a 3D Gaussian kernel. (e) Positive responses indicating locations suitable for the given sitting pose (colored red). (f) Adjusted pose at the location with the highest positive response.

above, f is focal length and r_{32} is a specific parameter in camera extrinsic matrix. A detailed derivation is available in the appendix. Fig. 2(f) illustrates the mapping process. We take the resulting pose depth as the depth of the pelvis joint and calculate the depths of other joints by their offsets w.r.t. the pelvis joint. Then, we map each joint into the 3D world using intrinsic and extrinsic camera matrices.

3.3. Affordance Constraint in the 3D World

Since the pose prediction model is trained with only 2D information, a plausible generated pose may not be physically feasible when mapped to 3D, e.g., the pose collides with the bed as exemplified in Fig. 2(d). Therefore, we adjust it locally to make the pose physically feasible. For example, we can adjust pose locations to avoid collision as shown in Fig. 2(e), or adjust a sitting pose right onto the surface of a bed as shown in Fig. 3(f).

The method by Gupta et al. [8] manually associates each action with an exemplar pose and searches locations valid for the pose by satisfying the *free space constraint* and the *support constraint*. However, such a manual solution is not feasible in our case since our poses are generated, rather than selected from a set of fixed poses. We explain next how to extend the method in [8] to search for locations satisfying both constraints in an efficient and fully-automatic manner.

Free space constraint. The free space constraint states that no human body parts can intersect with any object in the scene, such as furniture or walls. To satisfy this constraint, we perform a 3D correlation between poses and a voxel representation of the scene. We denote the voxelized 3D pose as p, with all voxel valued as one. We binarized the original voxel (Fig. 3(c)) with the free space as zero, and the occupied ones as one, denoted as V_f. The free space constraint is satisfied in the locations where R_f below a threshold T_f:

$$R_f = p * V_f \tag{1}$$

where $*$ indicates a 3D correlation operation. Necessary contacts between human and objects (e.g., a human touches the chair when sitting, or the floor when standing) should be considered. Thus we mask out these body parts that have to contact with objects, including thigh and pelvis for sitting poses and feet for standing poses, when performing the 3D correlation.

Support constraint. The support constraint states that the human pose should be supported by a surface of surrounding objects (e.g., floor, bed). We search locations that satisfy this constraint by performing two 3D correlations. The first correlation is performed between scene voxels V_s and a 3D Gaussian kernel to detect voxel cells on the surfaces of affordable objects (e.g., the bed in Fig. 3(d)). The V_s is produced by marking all voxels of affordable objects (chair, sofa, floor etc.) to zero, and the other voxels (including unoccupied voxels or objects that can not support a human pose) to one. After correlating with a 3D Gaussian kernel, all voxels except voxels on the boundaries will be either zero or one. Masking them out would leave us only voxels on affordable objects boundaries. We further mask out boundary voxels that do not have an upward surface normal.

Next, we perform another 3D correlation between poses and the object surfaces (see Fig. 3(e)) and take the location with the maximum correlation score as the optimal location for putting the pose (see Fig. 3(f)). Similar to the free space constraint discussed above, we denote the voxelized 3D human pose and pre-processed affordable object boundary voxel as p and V_s, the Gaussian kernel as G, then the *support* constraint R_s can be expressed as:

$$R_s = p * (G * V_s) \tag{2}$$

We adjust a pose to the "best location" where the person can comfortably lay or sit with maximal contacting area with the support surface. The location can be explicitly obtained through localizing at the point with $\max(R_s)$. Note that poses are adjusted in a local region to preserve the semantic information. Poses that do not find a valid location are discarded, i.e., the support constraint is satisfied in the locations where $\max(R_s)$ is above a threshold T_s.

4. 3D Affordance Generative Model

In this section, we show how to generate 3D human poses conditioned on a single scene image using the synthesized data described in Section 3. Generating human poses in 3D scenes requires modeling the joint distribution of human *scale*, *pose*, *location* and *interactions* with objects in 3D, which is very challenging. A typical solution is to use a single network to model the joint distribution of pose locations and gestures. This approach, however, will result in a huge solution space and poor performance, as analysed in Section 5.3. In contrast, we break it down to two jointly learned sub-tasks, where the generative model for each subtask is much easier to learn, To be specific, we first predict the plausible locations in a scene (see the *where* module in Fig. 1 and Fig. 4 (a)) and then predict the suitable human

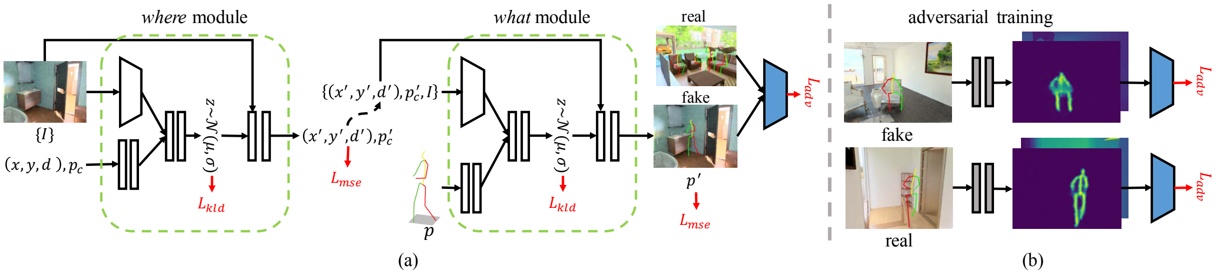


Figure 4. **Overview of the 3D affordance learning model.** (a) Our end-to-end framework consists of a *where* (Section 4.1) and a *what* (Section 4.2) component for pose location and gesture prediction respectively. (b) Detailed illustration of our adversarial training (blue block in (a), detailed in Section 4.3). Grey blocks convert joint coordinates and depth to a "depth heat map", which are pretrained and fixed when jointly training the *where* and the *what* module. Blocks with same color share parameters.

poses that are aligned with their surrounding context (see the *what* module in Fig. 1 and Fig. 4 (a)) of the predicted locations. Both modules are jointly trained using the pose location as a differentiable link, which allows the two modules to mutually benefit from each other, as well as from the discriminator described in Section 4.3.

We take two factors into consideration when designing both the *where* and the *what* modules. First, both modules should be able to understand the semantics of scene context to generate poses that follow natural human behaviors (e.g., sit rather than stand on a sofa). To this end, we model the distributions of pose locations and gestures by two VAEs conditioned on the scene context. We explain them in detail in Section 4.1 and Section 4.2 respectively. Second, both modules should be able to hallucinate 3D geometry of the scene to generate poses that obey physical rules in a scene (e.g., poses should be well supported by objects rather than float in the air). To achieve this goal, we introduce a geometry-aware discriminator that further regularizes the two modules to generate physically correct poses, which we discuss in Section 4.3. Fig. 4 illustrates the complete pipeline of our pose prediction model.

4.1. The *Where* Module: Pose Locations Prediction

Given a scene image I, we build a *where* VAE to encode pose location in the 3D scene, by simultaneously reconstructing pose pelvis joint coordinates (x, y) and depth d, as well as the most likely pose class p_c at the predicted location. The standard variational equality is represented as:

$$\log P(Y|I) - KL(Q(z|Y,I)||P(z|Y,I)) \tag{3}$$
$$= E_{z\sim Q}(logP(Y|z,I)) - KL(Q(z|Y,I)||P(z|I))$$

$P(z|I)$ and $Q(z|Y, I)$ are two normal distributions $\mathcal{N}(0, 1)$ and $\mathcal{N}(\mu(Y, I), \sigma(Y, I))$ and KL represents the Kullback-Leibler divergence.

The pose class p_c provides a clue for the likely pose appearance (e.g., sitting or standing), which can be obtained by assigning each pose to one of the 30 pose clusters described in [27]. Note that [27] uses an one-hot vector to represent the pose class, which does not consider the similarities of different pose typologies between classes. Here we directly represent p_c by the normalized center pose of each cluster so that similar pose classes also have similar representations, i.e., each $p_c \in \mathcal{R}^{3\times17}$ (each pose contains 17 joints).

The structure of the *where* module. As illustrated in Fig. 4 (a), the encoder extracts image features using an 18 layer ResNet [9] and concatenates them with the location features and pose class features extracted by two fully connected layers. The final concatenated feature is then fed into four fully connected layers to predict $\mu(Y, I)$ and $\sigma(Y, I)$ for distribution Q. The decoder takes a latent variable z sampled from Q and the scene context features shared with the encoder to predict $\{x, y, d, p_c\}$. Because it is challenging for the model to associate numerical coordinates with the exact location in the image, we predict a heat map in the decoder to indicate possible locations for a pose and adopt one Differentiable Spatial to Numerical Transform (DSNT) [20] layer to convert the heat map to pose location coordinates.

The objectives of the *where* module. We use three losses in training the *where* module. First, we minimize the Euclidean distance on the estimated pose class, depth and pelvis coordinates by $L_{mse} = \|Y^* - Y\|$. Second, we minimize the KL-divergence between the estimated distribution Q and the normal distribution $\mathcal{N}(0, 1)$ by $L_{kld} = KL[Q(z|\mu(Y, I), \sigma(Y, I))||\mathcal{N}(0, 1)]$. In addition, to better associate predicted pelvis joint depth and pixel coordinates, we minimize the Euclidean distance between ground truth and predicted pelvis coordinates under the world coordinate system using camera parameters for each scene. We refer this loss as *geometry loss* and represent it as $L_{geo} = \|M_e M_i[x^*, y^*, d^*] - M_e M_i[x, y, d]\|$, where M_e and M_i are camera extrinsic and intrinsic matrices. Our final objective is:

$$L = \lambda_{mse}L_{mse} + \lambda_{kld}L_{kld} + \lambda_{geo}L_{geo}, \tag{4}$$

where λ_{mse}, λ_{kld}, λ_{geo} are the weights that balance the

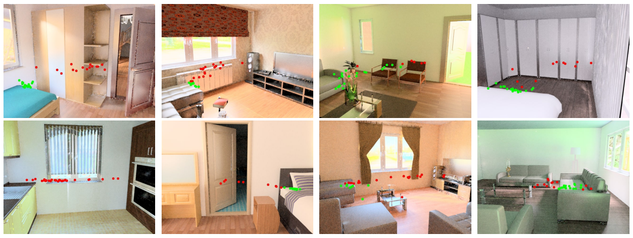


Figure 5. **Sampled locations by the "where" module.** In each scene, we show 30 locations sampled by our "where" module. For visualization purpose, we color pelvis joint locations for standing poses and sitting poses red and green respectively.

three objective terms.

We visualize the sampled locations conditioned on each scene image in Fig. 5. As shown in this figure, our "where" module (a) understands the scene and predicts reasonable locations for sitting poses around an affordable object or locations on correct height for standing poses. (b) generates multiple locations given a single scene image.

4.2. The *What* Module: Pose Gestures Prediction

The *what* module takes pelvis joint coordinates (x, y), depth d and pose class p_c predicted by the *where* module as well as a scene image I as inputs, and learns to predict coordinates and depth of each joint in $p \in \mathbb{R}^{3\times 17}$, so that the generated pose p can align well with its surrounding context. In other words, the *what* module needs to understand the scene context, and be able to sample poses conditioned on it. Similarly, we model the pose appearance distribution with a conditional VAE, which is represented as:

$$\begin{aligned} &\log(P(S|R,I)) - KL(Q(z|S,R,I)||P(z|S,R,I)) \\ &= E_{z\sim Q}(logP(P|z,R,I)) - KL(Q(z|S,R,I)||P(z|R,I)), \end{aligned} \tag{5}$$

where S represents the coordinates and depth $\{x, y, d\}$ for each joint, R denotes $\{x, y, d, p_c\}$ predicted by the *where* module. Other symbols follow those in Section 4.1.

The structure and objectives of the *what* module. Our *what* module shares similar structure as the *where* module (Fig. 4), except that the inputs are pose location, scene context and pose class, and the outputs are the coordinates and depth for each joint.

Similar to the *where* module, the *what* module contains three losses: a Euclidean loss on estimated joint coordinates and depth $L_{mse} = \|S^* - S\|$, a KL-divergence loss $L_{kld} = KL[Q(z|\mu(R,I), \sigma(R,I))||\mathcal{N}(0,1)]$, and a geometry loss $L_{geo} = \big\|M_e M_i[x_j^*, y_j^*, d_j^*] - M_e M_i[x_j, y_j, d_j]\big\|$, where $[x_j, y_j, d_j]$ are pixel coordinates and depth for joint j. While our goal is to model the shape of poses through modeling the joint distribution of joints S, the final objective is same as in Equation 4.

4.3. The Geometry-Aware Discriminator

In this work, we aim to generate poses in 3D scenes that follow physical rules in the scene, which requires our model to properly hallucinate the 3D scene geometry merely from a 2D image. To this end, in addition to including the depth value of each pose during training, we introduce a geometry-aware discriminator that further regularizes the *where* and *what* module simultaneously to generate poses that obey geometry rules in the scene.

As shown in Fig. 4(b), the discriminator takes generated poses and scene depth images as inputs and learns to discriminate between geometrically feasible (real) *vs.* unfeasible (fake) pairs. However, it is challenging for the discriminator to associate the discrete depth value of each joint to a scene depth map (i.e., the depth of each point between two connected joints is not modeled). Thus we first train a network which converts coordinates and depth of each joint to a "depth heat map" (Fig. 4(b)), where each pixel is either the depth of a point between two joints or -1 for background pixels. Details about the network are available in the appendix. We then feed this "depth heat map" together with the scene depth image into the discriminator. Our final adversarial objective is:

$$\begin{aligned} L_{adv}(G, D) = &\mathbb{E}_{c,p_r}[\log D(F(p_r), c)] + \\ &\mathbb{E}_{c,p_z}[\log(1 - D(F(p_z), c))] \end{aligned} \tag{6}$$

where G and D represent the pose prediction model and the discriminator model, F represents a pre-trained CNN that converts joint coordinates and depth to the "depth heat map" described above, p_r and p_z denote ground truth and generated poses, c denotes the depth image of the scene.

We note that both the geometry-aware discriminator as well as the geometrically feasible/unfeasible labels are utilized only during training. During testing, only the the part shown in Fig. 4(a) is needed to support single image conditioned generation, which makes the algorithm easy to be adapted to many application scenarios.

5. Experimental Results

In this section, we first introduce the details of our synthesized dataset and the quantitative evaluation metrics in Section 5.1. Then, we present the experimental results of our affordance prediction model in Section 5.2, as well as the ablation studies to understand how the main modules of the proposed algorithm contribute in Section 5.3. Finally, we compare the proposed method with the state-of-the-art affordance prediction method [27] in Section 5.4.

5.1. Dataset Synthesis and Evaluation Metrics

Dataset synthesis. As described in Section 3, we use the Sitcom dataset [27] for pose prediction in images and map the generated poses into the scene voxels in the SUNCG dataset [30, 26] for 3D pose affordance correction. In total, we apply the synthesizer to generate 1.5 million poses in $13,774$ SUNCG scenes. We use $13,074$ scenes for training and 700 scenes for evaluation.

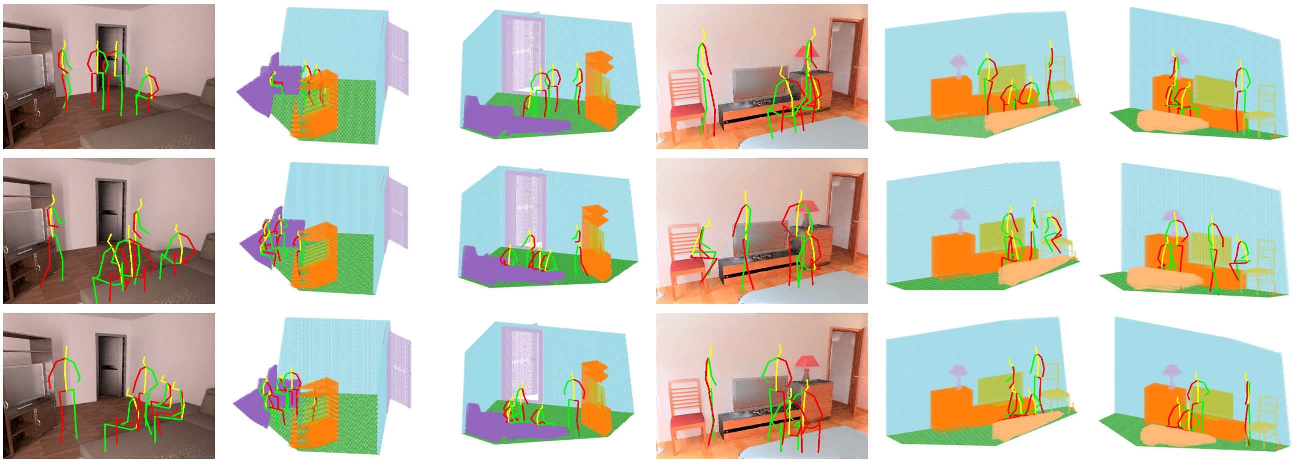
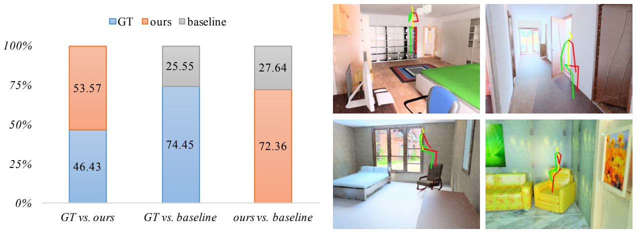


Figure 6. **Generated poses by our model.** The three rows show generated poses by models that take a RGB, RGB-D or depth map as input. For each scene, the first column illustrates pose projections in 2D scene images, and the last two columns illustrate poses in scene voxels visualized from different views.



(a) user study

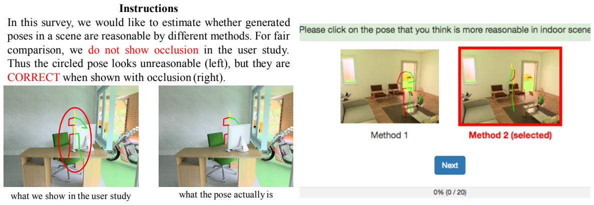
(b) negative samples

Figure 7. **Semantic plausibility evaluation.** (a) User study results. Each subject is asked to select the more reasonable pose through pairwise comparisons. The number indicates the percentage of preference on that comparison pair. "GT" means ground truth poses. (b) Manually annotated negative pose samples that are either impossible (column 1) or uncommon (column 2) in an indoor environment.

Quantitative evaluation metrics. The primary goal of this paper is to model 3D human affordance by generating human poses that are *semantically plausible* and *physically feasible* in a given scene. The semantic plausibility describes how reasonable a generated pose looks in an indoor environment. We design two ways to evaluate it.

First, we train a *pose authenticity classifier* to determine whether a generated pose is plausible. To train the classifier, we collect the ground truth poses from our synthesizer in Section 3 as positive samples, and manually annotate the negative samples following [27]. As shown in Fig. 7(b), the negative pose samples are either impossible or uncommon to appear in an indoor environment. In total, we collect $18,000$ pose samples in different scenes for training, and $1,400$ pose samples for evaluation. Both the training and the testing dataset contain an equal number of positive and negative poses. Our trained *pose authenticity classifier* achieves a classification accuracy as high as 86% on the testing dataset, and is ready to be used to test the plausibility of a pose, i.e., to check if a pose looks like a natural human pose in the given scene context. We define the ratio of poses that are classified as positive by the *pose authenticity classifier* as "semantic score". High semantic scores indicate that the model is able to understand the scene semantics to generate plausible poses in an indoor environment.

Second, we conduct a user study to let humans to determine how authentic the generated poses look like. Given a pair of poses sampled from ground truth poses and generated poses, either by the baseline method [27] or our method, in the same scene, a user is asked to select the pose that is more reasonable in an indoor environment. Fig. 8 shows the instructions and web UI. Note that since we focus on visual plausibility, both the generated/ground truth poses and the scenes for user study are projected and displayed as 2D images, which can be compared with [27].



Instructions

In this survey, we would like to estimate whether generated poses in a scene are reasonable by different methods. For fair comparison, we do not show occlusion in the user study. Thus the circled pose looks unreasonable (left), but they are CORRECT when shown with occlusion (right).

what we show in the user study

what the pose actually is

Please click on the pose that you think is more reasonable in indoor scene

Figure 8. **User study instructions and interface.** (Left) Instructions, (Right) User interface.

Finally, to check if a generated pose violates the geometric rules in a scene, we map it into the corresponding scene voxel, and check if the pose satisfies the free space constraint and support constraint as discussed in Section 3.3. We re-utilize the constraints as our evaluation criteria, by defining the ratio of poses that satisfy both constraints as *geometry score*. To be specific, for a standing pose, it satisfies the support constraint if the feet of the pose is within 8 voxel units (each voxel unit is 0.02 meter) of the floor. For a sitting pose, it satisfies the support constraint if there is an affordable surface (with $T_s >= 100$ as discussed in Section 3.3) within 8 voxel units of the pose. Furthermore, a pose that intersects less than or equal to 5 voxels (i.e. $T_f <= 5$) is considered satisfying the free space constraint. High geometry scores indicate that the model can hallucinate the 3D geometry and obey the geometry rules in the scene.

5.2. 3D Affordance Prediction

We visualize the generated poses by our *where* and *what* module with different input modalities in Fig. 6. We present quantitative evaluations in Table 1. For each model, we generate $3,500$ poses and calculate the semantic as well as geometry score over these poses. Note that the previous work [27] only focuses on predicting pose gestures at given locations. For a fair comparison, we combine the location heat map prediction model introduced in section 3.1, with the pose generator from [27] as our baseline model. Furthermore, since the baseline model is not able to predict the pose depth values, to calculate the geometry score described in Section 5.1, we adopt the strategy as introduced in Section 3.2 to estimate the pose depth and map the poses into the 3D scene.

Even with a single RGB image as input, our method achieves 19.47% higher semantic score, and 19.02% higher geometry score than the baseline model (see Table 1(b) and (c)). The results indicate that our model is able to understand both the context and moreover, the geometry of a scene. In addition, we generate 50 poses in different scenes and conduct the user study discussed in Section 5.1. In total, we collect 400 votes from 20 users and present the result in Fig. 7(a). According to the user study result, the poses generated by our method are not only more reasonable than poses predicted by the baseline method, but also indistinguishable from the ground truth poses.

Furthermore, we show that our pose prediction model can be further improved by including depth information of the scene. Specifically, we train two variants of our model that take a RGB-D or a depth map as input and present their performance in Table 1. From this table, we can see that including depth information of the scene constantly improve the geometry score of the pose prediction model under different experimental settings. Similar observations can also be found in Fig. 6, where the sitting pose generated by the model that takes a RGB image as input floats above the sofa (column 3, row 1), while the sitting pose generated by the model that takes a RGB-D or depth map as input aligns well with the sofa (column 3, row 2 and 3).

5.3. Ablation Studies

A single model for affordance learning. We conduct a baseline method to show that a single, straightforward generative network does not work for modeling complex joint distributions – we use a single VAE to encode 2D scene, pose locations and gestures, where all the other settings remain the same. We obtain semantic and geometry scores of 76.23 and 52.94 when taking RGB images as inputs (Table 1 (b)), which are much worse than the proposed method (Table 1 (c)).

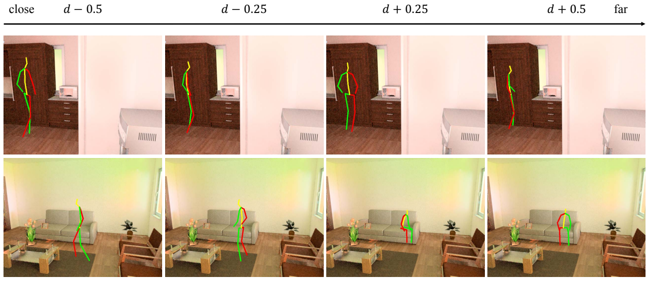


Figure 9. **Depth interpolation.** In the first scene (top row), closer poses are smaller than farther poses. In the second scene (bottom row), the generated poses change from standing to sitting when depth varies from $d-0.5$ to $d+0.5$, which corresponds well with the sofa in the back of the scene.

Joint training. First, we evaluate our model without joint training the *where* and *what* module. Table 1(c) *vs.* (e) shows the significant contribution of joint training for the semantic score. Without it, the semantic score reduces by 8.06% when taking a RGB image as input. We observe that although the model without joint training present higher geometry score, many of the generated locations have wrong depth values, which lead to unreasonably small poses that do not collide with other objects.

Adversarial training. Hallucinating 3D geometry purely based on 2D information is a challenging task. Thus we propose to use a geometry-aware discriminator which conditions on the depth map of a scene and learns to discriminate generated poses from "ground truth" poses (see Section 4.3). Table 1(c) *vs.* (d) shows the effectiveness of adversarial training. With adversarial training, our model is able to generate poses that better obey the rules of geometry in a scene (higher geometry score).

Geometry loss. A pose that looks plausible in a 2D context may still violate the rules of geometry when mapped into the 3D scene. Thus, to encourage our model to generate poses that are consistent with the geometry of the 3D world, we minimize the Euclidean distance between predicted poses and ground truth poses in the world coordinate space. Table 1(c) *vs.* (f) demonstrates the contribution of the geometry loss. Without it, the geometry score drops by 4.59% when taking a RGB image as input.

5.4. Comparison with State-of-the-Art

In this section, we follow the experimental settings by Wang et al. [27] and only focus on pose generation at given locations, i.e., the *what* module. To have a fair comparison, we train a *what* module that takes the same inputs as [27], i.e., the *2D pelvis coordinates* (x, y) and predicts

Table 1. **Quantitative evaluation of our affordance prediction model.** We show comparisons of our model with three different input modalities against the baseline model described in Section 5.2 in (b) and (c). Additionally, we show the performance of different variants of our model in (d) to (f) as discussed in Section 5.3.

(a) Metric	(b) Baseline	(c) Ours			(d) Ours w/o adversarial			(e) Ours w/o joint training			(f) Ours w/o geometry loss		
		RGB	RGB-D	Depth	RGB	RGB-D	Depth	RGB	RGB-D	Depth	RGB	RGB-D	Depth
semantic score	72.53	**91.69**	91.14	89.86	90.17	91.6	89.31	83.34	81.40	77.09	89.74	88.40	88.11
geometry score	23.25	66.40	71.17	**72.11**	62.71	72.00	70.91	46.46	71.37	60.83	56.11	66.40	63.77

Table 2. **Quantitative evaluation of the *what* module.** We show comparisons between the baseline model [27] and our model with three different input modalities.

Model	Baseline	Ours		
		RGB	RGB-D	Depth
semantic score	91.29	91.43	**91.86**	90.86
geometry score	56.29	78.43	82.00	**84.00**

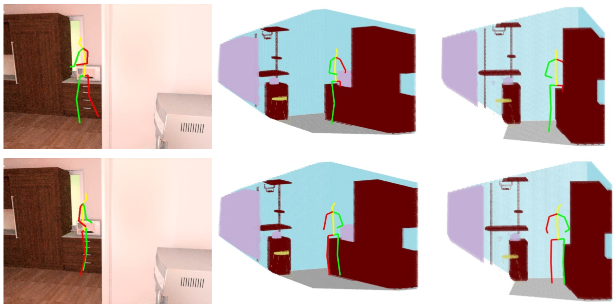


Figure 10. **Pose generation at given locations.** We show poses generated by the baseline method [27] (top row) and our method (bottom row) at given locations. The first column shows pose projections in scene images, and the last two columns show generated poses in 3D voxels visualized from two different views.

the coordinates as well as depth for each joint. We train the model in [27] on the SUNCG dataset with the synthesized poses for the ease of comparison. This model takes the *2D pelvis coordinates* (x, y) as our model but only predicts 2D coordinates of each joint. Table 2 shows the quantitative scores of these two models. Note that we use similar method to calculate geometry score for the baseline method discussed in Section 5.2. As shown in the table, our model achieves 6.66% higher geometry score, indicating that our model performs favorably in generating poses that obey the physical rules in the scene. The same observation can also be found in Fig. 10. Though given the same location, both the poses generated by our model and the baseline model appear plausible in the 2D image, only our generated pose is geometrically valid when mapped into the 3D scene.

A 2D coordinate (x, y) in a 2D scene image may correspond to multiple locations in the 3D scene with different depth values. A model that is able to hallucinate the geometry of a scene should be able to predict different poses at the same (x, y) location with different depth values. To inspect whether such geometry knowledge has been learned by our *what* module properly, we train another model that only depends on 3D pose locations and scene images. We particularly remove the pose class p_c in order to eliminate any clue that may indicate the geometrical information. Other settings are the same as the *what* model described in Section 5.2. During testing, we fix pelvis coordinates and the input scene image while interpolating depth between $d - 0.5$ to $d + 0.5$, where d is the ground truth pelvis depth. As we can see in Fig. 9, our model is able to generate poses with different scales and actions that well align with the scene according to different depth values, indicating its ability to hallucinate the 3D geometry of a scene properly.

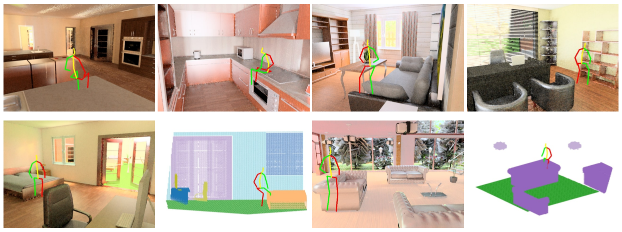


Figure 11. **Failure cases.** (Top) Semantic failure: failing to predict socially acceptable poses. (Bottom) Geometric failure: incorrectly hallucinating geometric information of the scene.

5.5. Failure Cases

Fig. 11 shows some failure cases. We mainly have two types of failure cases: (a) generated poses do not align well with the semantic context due to wrong semantic understanding of the scene (e.g., mistakenly sitting on the cabinet) (b) generated poses do not obey geometric rules (e.g., colliding with the objects in a scene). These are caused by a failure of object functionality understanding or 3D geometry hallucination based on 2D information, i.e., *reasoning*, which is an interesting open problem for future research.

6. Conclusion

In this work, we propose to predict *where* and *what* human poses can be put in 3D scenes using a two stage pipeline. We develop a 3D pose synthesizer that can produce millions of ground truth poses in 3D scenes automatically by fusing semantic and geometric knowledge from the Sitcom dataset [27] and a 3D scene dataset [26, 30]. Then we learn an end-to-end generative model that predicts both locations and gestures of human poses that are semantically plausible and geometrically feasible. Experimental results demonstrate the effectiveness of our proposed method against the stage-of-the-art human affordance prediction method.

Acknowledgement

We thank Soumyadip Sengupta and Jinwei Gu for providing the SUNCG-PBR dataset.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 2018. 2

[2] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler. Learning to act properly: Predicting and explaining affordances from images. In *CVPR*, 2018. 2

[3] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single view geometry. *IJCV*, 2014. 1, 2

[4] D. F. Fouhey, X. Wang, and A. Gupta. In defense of the direct perception of affordances. In *arXiv*, 2015. 1

[5] J. J. Gibson. The ecological approach to visual perception. *Houghton Mifflin*, 1979. 1, 2

[6] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object intaractions. *CVPR*, 2018. 2

[7] H. Grabner, J. Gall, and L. V. Gool. What makes a chair a chair? In *CVPR*, 2011. 1, 2

[8] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 2, 4

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 11

[10] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[11] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, jul 2014. 3, 11

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2014. 11

[13] D. Kinga and J. B. Adam. A method for stochastic optimization. In *ICLR*, 2015. 11, 12

[14] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013. 2

[15] H. S. Koppula and A. Saxena. Physically grounded spatio-temporal object affordances. In *ECCV*, 2014. 2

[16] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, 2018. 2, 12

[17] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *CVPR*, 2018. 2

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[19] A. Mallya and S. Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *ECCV*, 2016. 2

[20] A. Nibali, Z. He, S. Morgan, and L. Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018. 5

[21] X. Ouyang, Y. Cheng, Y. Jiang, C.-L. Li, and P. Zhou. Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond. *arXiv preprint arXiv:1804.02047*, 2018. 2

[22] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018. 3

[23] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *CVPR*, 2018. 1

[24] A. Roy and S. Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *ECCV*, 2016. 2, 3

[25] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. *Arxiv*, 2019. 12

[26] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2, 6, 9, 11

[27] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7, 8, 9, 11, 12

[28] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2

[29] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 2

[30] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *CVPR*, 2017. 2, 6, 9, 11

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[32] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*, 2013. 2

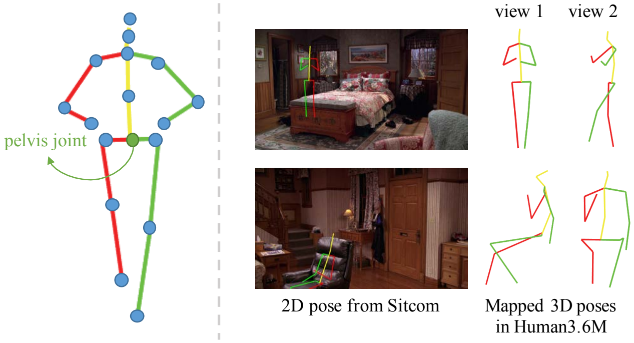
[33] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2

[34] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about Object Affordances in a Knowledge Base Representation. In *ECCV*, 2014. 2

[35] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu. Inferring forces and learning human utilities from videos. In *CVPR*, 2016. 1, 2

[36] Y. Zhu, Y. Zhao, and S.-C. Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, 2015. 2

A. From 2D Pose to 3D Pose (Section 3.1)



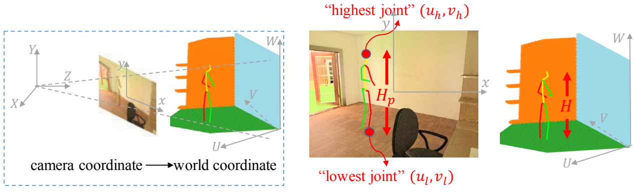
(a) (b)

Figure 12. (a) We represent each pose using the coordinates of 17 joints. We represent the pose location using the coordinates of the pelvis joint (green dot). (b) We map each 2D pose from [27] to a 3D pose in the Human3.6M dataset [12]. The last two columns show corresponding 3D pose of the 2D pose in column 1 visualized from two different views.

As discussed in Section 3.1, we map 2D poses annotated by Wang et al. [27] to 3D poses in the Human3.6M dataset [11]. This is carried out by first rotating each 3D pose by θ radian uniformly sampled from $[-\pi, \pi]$, then projecting it onto the xy plane. For each 2D pose, we search for its nearest neighbor with minimal Euclidean distance among all projected 2D poses and take the corresponding 3D pose as its 3D mapping. Fig. 12(b) shows examples of mapped 3D poses of 2D poses.

In this work, we represent pose location using the pelvis joint coordinates as shown in Fig. 12(a). For pose gesture representation, we use 17 joint coordinates, resulting a 34 dimensional vector for 2D poses and a 51 dimensional vector for 3D poses.

B. Mapping Poses into 3D Scenes (Section 3.2)


"highest joint" (u_h, v_h)
"lowest joint" (u_l, v_l)
H_p
H
camera coordinate ⟶ world coordinate
X
Y
Z
x
y
U
V
W

(a) mapping from image to voxel (b) camera coordinate (c) world coordinate

Figure 13. (a) Mapping from pixel coordinate system to world coordinate system. (b) Illustrations of human height H_p in pixel coordinates, "highest joint" and "lowest joint" in the pose. (c) Illustrations of human height in world coordinate system.

We present more details of how to estimate the depth of a pose (denoted as d), given the generated human pose on the image and the approximated human height in the real world, as described in Section 3.2. We sample human height H in the real world from a Gaussian distribution, i.e., $\mathcal{N}(1.65, 0.1)$ for standing poses and $\mathcal{N}(1.20, 0.1)$ for sitting poses. We denote the 2D coordinates of the "highest joint" (usually the head joint) and the "lowest joint" (usually the one of the foot joint) as (u_h, v_h) and (u_l, v_l) as shown in Fig. 13. In addition, we denote camera intrinsic matrix M_i and extrinsic matrix M_e as:

$$M_i = \begin{pmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{pmatrix} \quad M_e = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \tag{7}$$

The world coordinates of joint (u_h, v_h) is calculated by:

$$\begin{pmatrix} X_h \\ Y_h \\ Z_h \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x_{ch} \\ y_{ch} \\ d \\ 1 \end{pmatrix} \tag{8}$$

where $(x_{ch}, y_{ch}, d, 1)$ is the camera coordinate of (u_h, v_h), which is calculated by:

$$x_{ch} = \frac{(u_h - o_x)d}{f}, \qquad y_{ch} = \frac{(v_h - o_y)d}{f} \tag{9}$$

From (8) we have $Z_h = r_{31}x_{cl} + r_{32}y_{cl} + r_{33}d + t_1$. Similarly, we have the Z coordinate $Z_l = r_{31}x_{cl} + r_{32}y_{cl} + r_{33}d + t_1$ to represent the "lowest joint". Given the human height H in real world, we have $H = Z_h - Z_l = r_{31}(x_{ch} - y_{cl}) + r_{32}(y_{ch} - y_{cl})$. By substituting (9) into this equation, we have $H = \frac{r_{31}d}{f}(u_h - u_l) + \frac{r_{32}d}{f}(v_h - v_l)$. Note that $(u_h - u_l)$ and $(v_h - v_l)$ are the pose height H_p and width W_p in the pixel coordinate system as shown in Figure 12(b), thus we can calculate pose depth by $d = \frac{H \times f}{r_{31} \times W_p + r_{32} \times H_p}$. Specifically, for the SUNCG dataset [30, 26], $r_{32} = 0$ for all scenes, we simplify the depth estimation equation above as $d = \frac{H \times f}{r_{32} \times H_p}$, as concluded in Section 3.2.

C. Location Prediction in 2D Scene Images (Section 3.1)

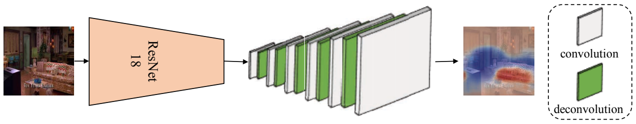


Figure 14. Architecture of the location prediction model. The encoder is a 18-layer ResNet [9] without the last two fully connected layers. The decoder is a 11-layer CNN with an extra Softmax layer at the end to normalized the generated heat map.

Fig. 14 illustrates the structure of our 2D pelvis location prediction model, as discussed in Section 3.1. Fig. 17 shows predicted heat maps and poses for the Sitcom [27] and the SUNCG dataset. We train the heat map prediction model for $5,000$ iterations using the Adam [13] solver. For data augmentation, we randomly crop a 384×384 patch from a 448×448 image, we set batch size to 100 and learning rate to 0.001. For pose generation at given locations, we use the same model as [27]. Note that instead of predicting 2D poses, we directly predict 3D poses obtained via 2D to 3D pose mapping, as described in Appendix A and Section 3.1.

D. More Details for 3D Pose Prediction (Section 4)

The *where* and *what* modules. We first train the *where* and *what* module discussed in Section 4.1 and 4.2 for 80000 iterations using the Adam [13] solver. Specifically, we set the batch size as 100 and the learning rate as 0.0001. Then, we connect the two modules and jointly finetune them with the geometry-aware discriminator, as introduced in Section 4.3, for another 50000 iterations. We adopt the similar training strategy as Lee et al. [16] and only use the discriminator to regularize an *unsupervised path* for both modules, i.e., the discriminator is used to regularize the distributions of generated poses that coming from the random noises, instead of interacting with the VAE block in a direct manner. We observe that such network architecture brings significant improvement to the generated results. Fig. 18 (a) shows the detailed structure of our *supervised* and *unsupervised* path and Fig. 18 (b), (c) shows the detailed structure of our *where* and *what* module.

Geometry-aware discriminator. As discussed in Section 4.3, we propose a geometry-aware discriminator to further regularize the generator to generate poses that obey the rules of geometry in a scene. However, it is challenging for the discriminator to associate joint coordinates, i.e., a 3-dimensional tensor, with the image. Therefore, we first train a CNN to convert the coordinates and depth of joints, into a "depth heat map" that has the same dimension as the input image. Fig. 18(d) illustrates the structure of this CNN. We train the CNN for 5000 iterations using the Adam [13] solver with a learning rate of 0.0002. Fig. 18(e) further shows the detailed structure of our geometry-aware discriminator.

E. Additional Experimental Results

We show **synthesized poses** in scene images and voxels in Fig. 15. More results of **generated poses** in images and scene voxels are shown in Fig. 16. Note that in this work we use the SUNCG-PBR dataset by Sengupta et al. [25]. Despite noise introduced by the rendering process, our pose prediction model is still able to predict plausible poses.

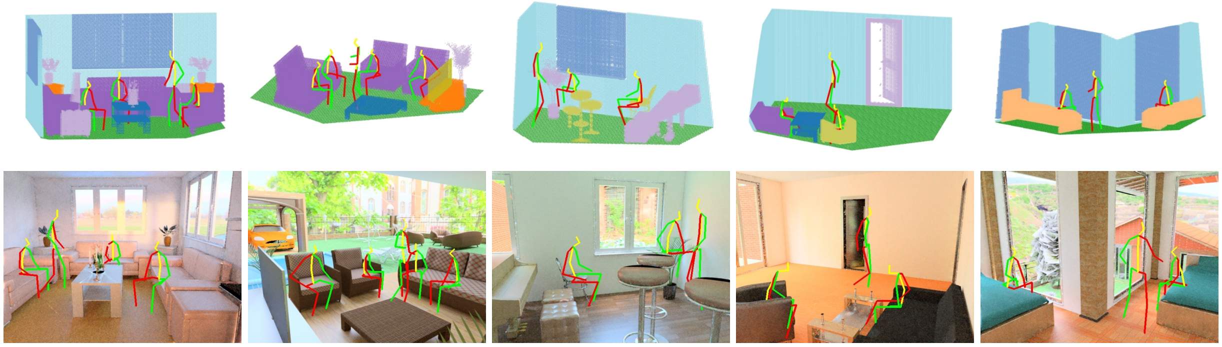


Figure 15. Samples of synthesized poses. (Top) Sample poses shown in 3D scene geometry, and (Bottom) rendered images of the corresponding scenes. Note that the generated poses contain information about occlusion in the scene.

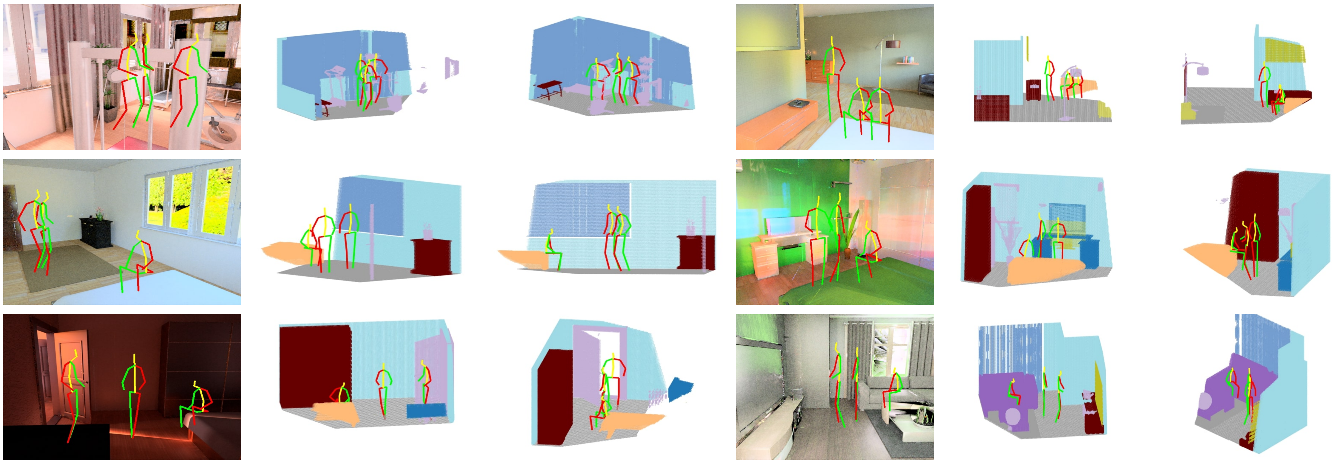


Figure 16. Generated poses by our pose prediction model. We show generated poses in images (first column) and voxels visualized from two different views (last two columns) for each scene. Poses are generated by our model which takes a single depth image as input.

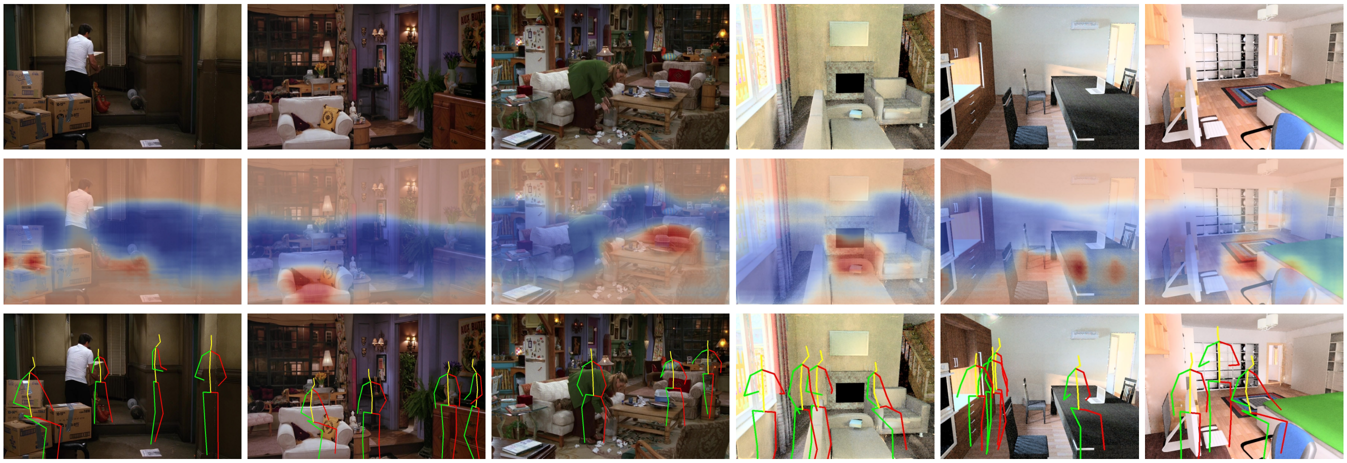
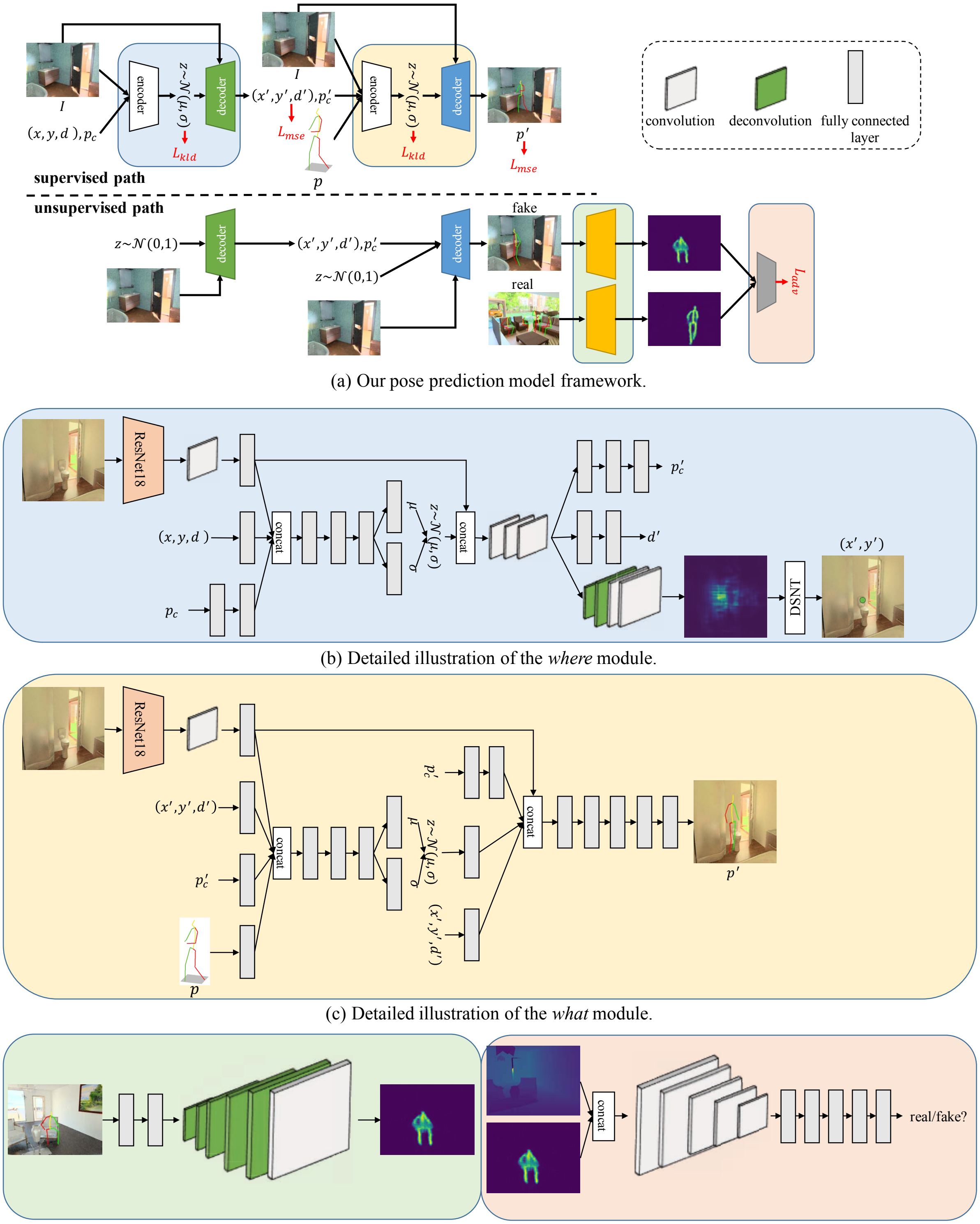


Figure 17. Predicted pose location heat maps and sampled poses. First three columns show results on Sitcom and last three columns show results on SUNCG. For visualization purpose, we summarize area suitable for sitting pose location (shown as red area), area suitable for standing pose location (shown as blue) as well as area not suitable for any human poses (shown as light yellow) in predicted heat maps. All poses shown in the bottom row are projections of 3D poses generated by our model.



(a) Our pose prediction model framework.

(b) Detailed illustration of the *where* module.

(c) Detailed illustration of the *what* module.

(d) The CNN converts the joint coordinates and depth to a "depth heat map".

(e) The structure of the geometry-aware discriminator.

Figure 18. Detailed structure of each block in our pose prediction model. We show the overview of our pose prediction model, including the *supervised* and *unsupervised* path explained in Appendix D in Fig. (a). Detailed structure of each block is illustrated in (b), (c), (d) and (e) respectively, same blocks are filled with same background color.