Fundamental Question

What is a good vector representation of an object?

● Something that can be easily predicted from 2D images?

● Something that allows for the generation of 3D models of objects?
Previous Work I (2D Image Representations)

- Olshausen and Field (1996): Objective was to obtain a representation that was sparse and could reconstruct image pixels.
- Hinton and Salakhutdinov (2006), Vincent et al. (2010), and Salakhutdinov and Hinton (2009): Use stacked RBMs or DBMs to learn a low-dimensional representation for reconstruction.
- Goodfellow et al. (2014): GANs

Focuses on 2D image representations.

No inclusion of 3D information explicitly.
Previous Work II (3D Object Representations)

- Wu et al. (2015), Li et al. (2016), Maturana and Scherer (2015): Using deep architectures to represent and classify 3D objects

Focuses on 3D object representations

Neglects the mapping from 2D images to 3D objects
Previous Work III (Towards unifying 3D and 2D representations)

- Dosovitskiy et al. (2015): Use 3D CAD models to train a parameterized generative model for objects to learn a 3D-aware representation for 2D images.
- Kulkarni, Whitney, et al. (2015): Guide the latent representation in a generative model (for 2D images) to explicitly model certain 3D properties such as pose, light, shape, etc.

Focus on using 3D data to inform/guide 2D image representations.
This Work

Aims to unify the previous two lines of work and create a representation that is both predictable from 2D images and generative for 3D objects.
TL-Embedding Network
TL-Embedding Network (Autoencoder)

Parameterized ReLU throughout autoencoder

Normal ReLU  Parametric ReLU
TL-Embedding Network (Mapping Image to Embedding)

Train (T-Network)

AlexNet Architecture for the ConvNet

Rendered Chairs
TL-Embedding Network (Training)

Notice the two losses!
TL-Embedding Network (Testing)

Model Pipeline

2D Image -> Representation -> 3D Model
Why is this architecture better for 2D -> 3D data?

The explicit combination of both 2D and 3D data to create a unified representation vector is the dominant separating factor from others.

It explicitly focuses on the mapping of 2D to 3D data whereas other models focus on only 2D or only 3D representation.
Comparison to Previous Architectures (2D)

An architecture for explicitly representing 2D images

Comparison to Previous Architectures (3D)

An architecture for explicitly classifying 3D objects

Comparison to Previous Architectures (2D + 3D)

An architecture for generating 2D images with specific 3D properties.

Training Data Generation: (Voxel Grid, Image) Pairs

**Voxelized 3D models:** CAD models of beds, cabinets, chairs, sofas and tables from Stanford ShapeNet voxelized using the voxelizer from the Wu et al. (2015) 3D ShapeNets paper.

**2D Images of 3D models:** Render the CAD models into 72 views (3 elevations of 15°, 30° and 45°, 24 azimuth angles from 0 to 360°, in 15° increments). The resulting views are rendered on top of randomly selected open room images downloaded from the internet.
Training I (Autoencoder only)

Input: CAD model voxel grid --- Target: the same voxel grid as the input

Initialize the autoencoder with random weights from $N(0, 0.01)$

Train it end-to-end for ~200 epochs using the cross-entropy loss on the final voxel output against the original voxel input

$$E = -\frac{1}{N} \sum_{n=1}^{N} [p_n \log \hat{p}_n + (1 - p_n) \log(1 - \hat{p}_n)]$$

$N$ is $20^3$. Each $n$ is one voxel in the input and output voxel grids.
Training II (ConvNet only)

Input: 2D image of a rendered CAD model --- Target: the autoencoder’s 64D representation of the same CAD model’s voxel grid

Initialize the ConvNet with parameters trained on ImageNet

Train the ConvNet to regress to the 64D autoencoder representation via the Euclidean loss, keeping the convolutional layers fixed.

\[ E = \| \hat{y}_d - y_d \|_2^2 \]

\( \hat{y}_d \) is the ConvNet’s 64D final layer, \( y \) is the autoencoder’s 64D representation
Training III (Joint)

Inputs: CAD model voxel grid and 2D image of the rendered CAD model ---
Targets: the same CAD model voxel grid and the autoencoder’s 64D representation of the same CAD model’s voxel grid

Scale the autoencoder loss to have approximately the same initial value as the ConvNet loss.

Train jointly!

Arrows show direction of gradient backprop from the two losses.
Experiment

Evaluate the representation:

1. Generative in 3D?
   
   64D representation $\rightarrow 20^3$ voxel

2. Predictable from 2D images?
   
   2D images $\rightarrow$ 64D representation
Experiment Aims

Evaluate the representation:

1. Generative in 3D?
2. Predictable from 2D images?

Steps:

1. Representation: reconstructive, smooth, class distinguishable
2. Predict voxels from 2D image
3. CAD retrieval from natural images
4. 3D shape arithmetic
Dataset

**ShapeNet[39] CAD Dataset:** rendered images with random background

5 kinds of common indoor objects: chair (6778 models), table (8509 models), sofa (3173 models), cabinet (1572 models), and bed (254 models).

16228 train and 4058 test objects

**IKEA Dataset[23]:** natural indoor images, cluttered background

937 cropped single object images labeled with 225 3D models.

Experiment Steps

1. **Autoencoder: reconstruction, smoothness, classification**
2. Predict voxels from 2D image
3. CAD retrieval from natural images
4. 3D shape arithmetic
1. **Reconstruction, Smoothness, Classification**
1. **Reconstruction, Smoothness, Classification**

- Even with **joint training** for more predictable representation from 2D images, still preserves the overall reconstruction performance.

<table>
<thead>
<tr>
<th>Table 1: Reconstruction performance using AP on test data.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Proposed (before Joint)</td>
</tr>
<tr>
<td>Proposed (after Joint)</td>
</tr>
<tr>
<td>PCA</td>
</tr>
</tbody>
</table>
1. Reconstruction, **Smoothness**, Classification

Linear interpolation between representations of randomly picked models (A & B)
Explored if the dimensions of representations are meaningful.

1. Reconstruction, **Smoothness**, Classification

First 32D of A

Latter 32D of B

New 64D latent representation
1. Reconstruction, **Smoothness**, Classification

Explored the representation dimensions’ **meanings**: scaling each dimension separately and analyzing the effect on the reconstruction

Higher values in dimension 22 lead to thicker legs | higher values in 9 lead to disappearance of legs.
1. Reconstruction, Smoothness, **Classification**

CAD 40-way classification:


<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>[39]</td>
<td>77.32%</td>
<td>Finetune the representation for classification</td>
</tr>
<tr>
<td>Ours</td>
<td>74.4%</td>
<td>Don’t finetune the representation for classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pairwise feature augmentation and trained a linear SVM</td>
</tr>
<tr>
<td>PCA</td>
<td>68.4%</td>
<td>Don’t finetune the representation for classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pairwise feature augmentation and trained a linear SVM</td>
</tr>
</tbody>
</table>

Experiment Steps

1. Autoencoder: reconstruction, smoothness, classification

2. Predict voxels from 2D image
2. Voxel prediction from 2D

Compared with Kar et al. [17]

Voxelize [17] output and ground truth, and compute the overlap P-R curve with alignment.

Maximum F-1 score: 49.2% (ours) to 46.3% [17].

Experiment Steps

1. Autoencoder: reconstruction, smoothness, classification
2. Predict voxels from 2D image
3. CAD retrieval from natural images
Experiment Steps

1. Autoencoder: reconstruction, smoothness, classification
2. Predict voxels from 2D image
3. CAD retrieval from natural images
4. **3D shape arithmetic**
Conclusion

Smooth, discriminant representations that simultaneously allow for 2D representation and 3D generation.

Limitations:

- Image with single object
- Small category of indoor furnitures
Related Work 1

Related Work 1

Related Work 1

Related Work 2

## Related Work 2

|-------|--------------|------|------------|-------|--------------|------|------------|
Related Work 3

Related Work 3

Thanks!

Questions?
Supplemental Slides
Related work further


Figure 7: Our approach identifies a variety of objects in an office scene.
2. Predict voxels from 2D image

Baseline

Direct prediction: removing autoencoder
ImageNet pre-trained AlexNet: 2D $\rightarrow$ $20^3$ voxels

1. Direct-conv4: freeze all layers before conv4
2. Direct-fc8: freeze all layers except fc8

Without Joint: without the joint fine-tuning (jointly 3D generative and 2D predictable)

1. Train the autoencoder independently
2. Train the ConvNet to regress to the 64D representation
2. Voxel prediction from 2D

Table 2: Average Precision for Voxel Prediction on the CAD test set. The Proposed TL-Network outperforms the baselines on each object.

<table>
<thead>
<tr>
<th></th>
<th>Chair</th>
<th>Table</th>
<th>Sofa</th>
<th>Cabinet</th>
<th>Bed</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed (with Joint)</td>
<td>66.9</td>
<td>59.7</td>
<td>79.3</td>
<td>79.3</td>
<td>41.9</td>
<td>65.4</td>
</tr>
<tr>
<td>Proposed (without Joint)</td>
<td>66.6</td>
<td>57.5</td>
<td>79.3</td>
<td>76.5</td>
<td>33.8</td>
<td>62.7</td>
</tr>
<tr>
<td>Direct-conv4</td>
<td>40.9</td>
<td>23.7</td>
<td>58.1</td>
<td>44.3</td>
<td>23.1</td>
<td>38.0</td>
</tr>
<tr>
<td>Direct-fc8</td>
<td>21.8</td>
<td>15.5</td>
<td>35.6</td>
<td>32.7</td>
<td>18.6</td>
<td>24.8</td>
</tr>
</tbody>
</table>

Table 3: Average Precision for Voxel Prediction on the IKEA dataset.

<table>
<thead>
<tr>
<th></th>
<th>Bed</th>
<th>Bookcase</th>
<th>Chair</th>
<th>Desk</th>
<th>Sofa</th>
<th>Table</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>56.3</td>
<td>30.2</td>
<td>32.9</td>
<td>25.8</td>
<td>71.7</td>
<td>23.3</td>
<td>38.3</td>
</tr>
<tr>
<td>Direct-conv4</td>
<td>38.2</td>
<td>26.6</td>
<td>31.4</td>
<td>26.6</td>
<td>69.3</td>
<td>19.1</td>
<td>31.1</td>
</tr>
<tr>
<td>Direct-fc8</td>
<td>29.5</td>
<td>17.3</td>
<td>20.4</td>
<td>19.7</td>
<td>38.8</td>
<td>16.0</td>
<td>19.8</td>
</tr>
</tbody>
</table>
3. CAD Retrieval: Baseline

Baseline:

Rendered all the 225 models at 30 deg. elevation and 8 uniformly sampled azimuths from 0 to 360 deg. onto a white background.

ImageNet trained AlexNet’s fc7 features over the query image and renderings to perform nearest neighbor search (cosine distance).

...strong baseline with much higher resolution of images, each 3D model using 8 vectors of 4096D (slower)

...our 20^3 voxel grids, uses only a single 64D
3. CAD Retrieval: Evaluation

Rank all 225 CAD models in the corpus by cosine distance.

(a) Instance match: at what rank does the exact-match CAD model appear?

(b) Category match: at what rank does the first model of the same category appear?

Table 4: Mean recall @10 of ground truth model in retrievals for our method and baseline described in Sec. 4.4

<table>
<thead>
<tr>
<th></th>
<th>Sofa</th>
<th>Chair</th>
<th>Bookcase</th>
<th>Bed</th>
<th>Table</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>32.3</td>
<td>41.0</td>
<td>26.8</td>
<td>38.5</td>
<td>8.0</td>
<td>29.3</td>
</tr>
<tr>
<td>Fc7-NN</td>
<td>14.6</td>
<td>33.9</td>
<td>23.5</td>
<td>7.7</td>
<td>17.4</td>
<td>19.4</td>
</tr>
</tbody>
</table>
3. CAD Retrieval
Conclusion

Smooth, discriminant representations that simultaneously allow for 2D representation and 3D generation:

- Train models that explicitly represent and reconstruct 3D objects (e.g. an autoencoder).
- Train models that explicitly learn a representation for a 2D image, the goal here is to mirror the representation learned via the 3D model.
- Jointly train the above models to force the two representations to match without compromising performance in their individual tasks.

Limitations:

- Single object on cropped images