Multiview Compressive Coding for 3D Reconstruction

Chao-Yuan Wu Justin Johnson Jitendra Malik Christoph Feichtenhofer Georgia Gkioxari
FAIR, Meta AI

Abstract

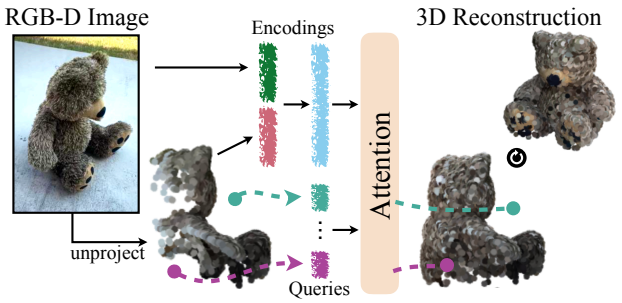
A central goal of visual recognition is to understand objects and scenes from a single image. 2D recognition has witnessed tremendous progress thanks to large-scale learning and general-purpose representations. Comparatively, 3D poses new challenges stemming from occlusions not depicted in the image. Prior works try to overcome these by inferring from multiple views or rely on scarce CAD models and category-specific priors which hinder scaling to novel settings. In this work, we explore single-view 3D reconstruction by learning generalizable representations inspired by advances in self-supervised learning. We introduce a simple framework that operates on 3D points of single objects or whole scenes coupled with category-agnostic large-scale training from diverse RGB-D videos. Our model, Multiview Compressive Coding (MCC), learns to compress the input appearance and geometry to predict the 3D structure by querying a 3D-aware decoder. MCC's generality and efficiency allow it to learn from large-scale and diverse data sources with strong generalization to novel objects imagined by DALL·E 2 or captured in-the-wild with an iPhone.

1. Introduction

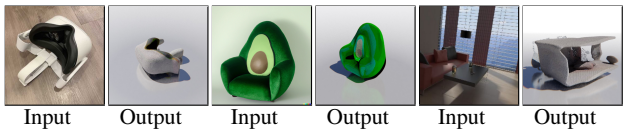
Images depict objects and scenes in diverse settings. Popular 2D visual tasks, such as object classification [8] and segmentation [33, 83], aim to recognize them on the image plane. But image planes do not capture scenes in their entirety. Consider Fig. 1a. The toy's left arm is not visible in the image. This is framed by the task of 3D reconstruction: given an image, *fully* reconstruct the scene in 3D.

3D reconstruction is a longstanding problem in AI with applications in robotics and AR/VR. Structure from Motion [19, 61] lifts images to 3D by triangulation. Recently, NeRF [38] optimizes radiance fields to synthesize novel views. These approaches require many views of the same scene during inference and do not generalize to novel scenes from a single image. Others [17, 68] predict 3D from a single image but rely on expensive CAD supervision [6, 60].

Project page: https://mcc3d.github.io



(a) **MCC Overview**



(b) **3D Reconstructions by MCC**

Figure 1. **Multiview Compressive Coding (MCC).** (a): MCC encodes an input RGB-D image and uses an attention-based model to predict the occupancy and color of query points to form the final 3D reconstruction. (b): MCC generalizes to novel objects captured with iPhones (left) or imagined by DALL·E 2 [48] (middle). It is also general – it works not only on objects but also scenes (right).

Reminiscent of generalized cylinders [41], some introduce object-specific priors via category-specific 3D templates [26, 30, 32], pose [43] or symmetries [73]. While impressive, these methods cannot scale as they rely on onerous 3D annotations and category-specific priors which are not generally true. Alas large-scale learning, which has shown promising generalization results for images [46] and language [3], is largely underexplored for 3D reconstruction.

Image-based recognition is entering a new era thanks to domain-agnostic architectures, like transformers [11, 65], and large-scale category-agnostic learning [20] . Motivated by these advances, we present a scalable, general-purpose model for 3D reconstruction from a single image. We introduce a simple, yet effective, framework that operates directly on 3D points. 3D points are general as they can capture any objects or scenes and are more versatile and efficient than meshes and voxels. Their generality and efficiency enables large-scale category-agnostic training. In turn, large-scale training makes our 3D model effective.

Central to our approach is an input encoding and a queriable 3D-aware decoder. The input to our model is a single RGB-D image, which returns the visible (*seen*) 3D points via unprojection. Image and points are encoded with transformers. A new 3D point, sampled from 3D space, queries a transformer decoder conditioned on the input to predict its occupancy and its color. The decoder reconstructs the full, *seen* and *unseen*, 3D geometry, as shown in Fig. 1a. Our occupancy-based formulation, introduced in [37], frames 3D reconstruction as a binary classification problem and removes constraints pertinent to specialized representations (*e.g.*, deformations of a 3D template) or a fixed resolution. Being tasked with predicting the *unseen* 3D geometry of diverse objects or scenes, our decoder learns a strong 3D representation. This finding directly connects to recent advances in image-based self-supervised learning and masked autoencoders (MAE) [20] which learn powerful image representations by predicting masked (unseen) image patches.

Our model inputs single RGB-D images, which are ubiquitous thanks to advances in hardware. Nowadays, depth sensors are found in iPhone's front and back cameras. We show results from iPhone captures in §4 and Fig. 1b. Our decoder predicts point cloud occupancies. Supervision is sourced from multiple RGB-D views, *e.g.*, video frames, with relative camera poses, *e.g.*, from COLMAP [55, 56]. The posed views produce 3D point clouds which serve as proxy ground truth. These point clouds are far from "perfect" as they are amenable to sensor and camera pose noise. However, we show that when used at scale they are sufficient for our model. This suggests that 3D annotations, which are expensive to acquire, can be replaced with many RGB-D video captures, which are much easier to collect.

We call our approach Multiview Compressive Coding (MCC), as it learns from many views, compresses appearance and geometry and learns a 3D-aware decoder. We demonstrate the generality of MCC by experimenting on six diverse data sources: CO3D [51], Hypersim [52], Taskonomy [81], ImageNet [8], in-the-wild iPhone captures and DALL·E 2 [48] generations. These datasets range from large-scale captures of more than 50 common object types, to holistic scenes, such as warehouses, auditoriums, lofts, restaurants, and imaginary objects. We compare to state-of-the-art methods, tailored for single objects [21, 51, 79] and scene reconstruction [31] and show our model's superiority in both settings with a unified architecture. Enabled by MCC's general purpose design, we show the impact of large-scale learning in terms of reconstruction quality and zero-shot generalization on novel object and scene types.

2. Related Work

Multiview 3D reconstruction is a longstanding problem in computer vision. Traditional techniques include binocular stereopsis [70], SfM [19,54,61–63], and SLAM [5,58]. Reconstruction by analysis [12] or synthesis via volume rendering [25] of implicit [38, 82] and explicit [34, 57] representations have shown to produce strong results. Supervised approaches predict voxels [67, 74] or meshes [69, 71] by training deep nets. These techniques produce high-quality outputs, but rely on multiple views at test time. In this work, we assume a single RGB-D image during inference.

Single-view 3D reconstruction is challenging. One line of work trains models that predict 3D geometry via CAD [17, 68], meshes [31,75], voxels [16,72] or point clouds [13,37] supervision. Results are commonly demonstrated on synthetic simplistic benchmarks, such as ShapeNet [6], or for a small set of object categories, as in Pix3D [60]. Weakly supervised approaches use category-specific priors via 3D shape templates [18, 26, 30] and pose [43] or learn via 2D silhouettes and re-projection on posed views [7, 27, 35, 50]. While impressive, these approaches are limited to specific objects from a closed-world vocabulary. Some [66, 77] explore category-agnostic models, but focus on synthetic datasets. In this work, we learn a general-purpose 3D representation from RGB-D views from a diverse and large set of data sources of real-world objects and scenes.

Shape completion methods complete the 3D geometry of partial reconstructions. For objects, methods directly output full point clouds [22, 79, 80] or deploy generative models [76, 84], but are typically tied to a fixed resolution. For scenes, techniques include plane fitting [39], 3D model fitting and retrieval [15, 40] or leverage symmetries [28] and predict 3D semantics [4, 14, 59]. Our model tackles both objects and scenes with a unified architecture and outputs any-resolution 3D geometry with a 3D-aware decoder. We compare to recent shape completion techniques.

Implicit 3D representations such as SDFs [44, 53] and occupancy nets (OccNets) [37] have proven effective 3D representations. NeRF [38] optimizes per-scene neural fields for view synthesis. NeRF extensions target scene generalization by encoding input views with deep nets [21, 47, 78] or improve reconstruction quality by supervising with depth [9]. MCC adopts an occupancy-based representation, similar to OccNets [37], with an attention mechanism on encoded appearance and geometric cues which allows it to predict in any 3D region, even outside the camera frustum, efficiently. We show that this strategy outperforms the global-feature strategy from OccNets [37] or single-location features used in NeRF-based methods [21, 51].

Self-supervised learning has advanced image [2, 20, 46] and language [3, 10] understanding. For images, masked autoencoders [20] paired with transformers and large-scale category-agnostic training learn general representations for 2D recognition. We draw from these findings and extend the architecture and learning for the task of 3D reconstruction.

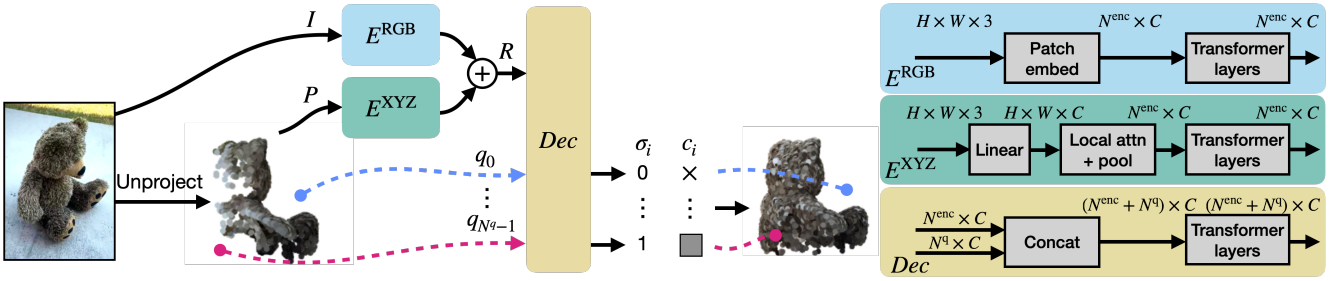


Figure 2. **Model Overview.** Given an RGB-D image, MCC unprojects the pixels of the input RGB image I to the corresponding 3D points P. An image encoder E^{RGB} and a geometry encoder E^{XYZ} encode I and P into a 3D-aware representation R. A decoder predicts the occupancy σ_i and color c_i of query q_i, $i = 0, \ldots, N_q - 1$, conditioned on R. The predicted colored points form the final 3D reconstruction.

3. Multiview Compressive Coding (MCC)

MCC adopts an encoder-decoder architecture. The input RGB-D image is fed to the encoder to produce encoding R. The decoder inputs a *query* 3D point $q_i \in \mathbb{R}^3$, along with R, to predict its occupancy probability $\sigma_i \in [0, 1]$, as in [37], and RGB color $c_i \in [0, 1]^3$. Fig. 2 illustrates our model.

During training, we supervise MCC with "true" points derived from posed RGB-D views. These point clouds serve as ground truth: q_i is labeled as positive if it is close to the ground truth and negative otherwise. Intuitively, the other views guide the model to reason about what parts of the *unseen* space belong to the object or scene. As a result, the input encoding R learns a representation of the *full* 3D geometry and guides the decoder to make the right prediction.

During inference, the model predicts occupancy and color for a grid of points at any desired resolution. The set of occupied colored points forms the final reconstruction.

MCC requires only *points* for supervision, extracted from posed RGB-D views, *e.g.*, video frames. Note that the derived point clouds, which serve as ground truth, are far from perfect due to noise in the captures and pose estimation. However, when used at scale they are sufficient. This deviates from OccNets [37] and other distance-based works [44, 53] which rely on clean CAD models or 3D meshes. This is an important finding as it suggests that expensive CAD supervision can be replaced with cheap RGB-D video captures. This property of MCC allows us to train on a wide range of diverse data. In §4, we show that large-scale training is crucial for high-quality reconstruction.

3.1. MCC Encoder

The input to our model is a single RGB-D image. Let $I \in \mathbb{R}^{H \times W \times 3}$ be the RGB image and $\Delta \in \mathbb{R}^{H \times W}$ the associated depth. We use Δ to unproject the pixels into their positions $P \in \mathbb{R}^{H \times W \times 3}$ in 3D. I and P are encoded into a single representation R via

$$R := f\left(E^{\text{RGB}}(I), E^{\text{XYZ}}(P)\right) \in \mathbb{R}^{N^{\text{enc}} \times C} \qquad (1)$$

E^{RGB} and E^{XYZ} are two transformers [65]. E^{RGB} follows a ViT architecture [11] to encode the input image I. E^{XYZ} processes the input points P similar to a ViT, but encodes 3D coordinates instead of RGB color channels. We explain in detail how to adapt a ViT to encode the input points P in §3.4. f concatenates the two outputs from the transformers along the channel dimension followed by a linear projection to C-dimensions. N^{enc} is the number of tokens used in the transformers. Fig. 2 shows an illustration.

The proposed two-tower design is general and performant. Alternative designs are ablated in §4.

3.2. MCC Decoder

The decoder takes as input the output of the encoder, R, and N^q 3D point queries q_i, $i = 0, \ldots, N_q - 1$, to predict occupancy and colors for each point,

$$(\sigma_0, c_0), (\sigma_1, c_1), \ldots := Dec(R, q_0, q_1, \ldots) \qquad (2)$$

The decoder Dec linearly projects each query q_i to C-dimensions (the same as R), concatenates them with R in the token dimension, and then uses a transformer to model the interactions between R and queries. We draw inspiration from MAE [20] for this design. The output feature of each query token is passed through a binary classification head that predicts its occupancy σ_i, and a 256-way classification head that predicts its RGB color c_i [64].

As described in Eq. 2, we feed multiple queries to the decoder for efficiency via parallelization, which significantly speeds up training and inference. However, since all tokens attend to all tokens in a standard transformer, this creates undesirable dependencies among queries. To break the unwanted dependencies, we mask out the attention weights such that tokens cannot attend to the other queries (except for self). This masking pattern is illustrated in Fig. 3.

MCC's attention architecture differentiates it from prior 3D reconstruction approaches. In [37, 42], points condition on a globally pooled image feature; in [21, 47, 78] they condition on the projected locations of the image feature map. In §4 we show that MCC's design performs better.

The computation of the decoder grows with the number of queries, while the encoder embeds the input image once regardless of the final output resolution. By using a relatively lightweight decoder, our inference is made efficient

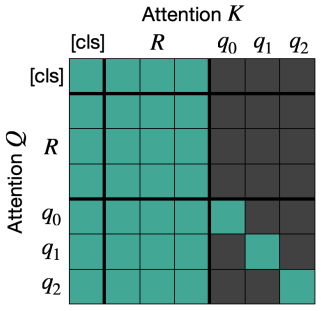


Figure 3. **Attention Masking Pattern in MCC's Decoder.** The masking in MCC's decoder ensures a query cannot depend on another, apart from itself. `cls` is a learnable global summary token, following [10, 11].

even at high resolutions, and the encoder cost is amortized. This allows us to dynamically change output resolutions and does not require re-computing the input encoding R.

3.3. Query Sampling

Training. MCC samples $N^q = 550$ queries from the 3D world space uniformly and per training example. We ablate sampling strategies in §4. A query is considered "occupied" (positive) if it is located within radius $\tau = 0.1$ to a ground truth point, and "unoccupied" (negative) otherwise. The ground truth is defined as the union of all unprojected points from all RGB-D views of the scene.

Inference. We uniformly sample a grid of points covering the 3D space. Queries with occupancy score greater than a threshold of 0.1 and their color predictions form the final reconstruction. Techniques such as Octree [36] could be easily integrated to further speed up test-time sampling.

3.4. Implementation Details

E^{XYZ} **Patch Embeddings.** Note that the depth values, and consequently the 3D locations in P, might be unknown for some points (*e.g.*, due to sensor uncertainty). Thus, the convolution-based patch embedding design in a ViT [11] is not directly applicable. We use a self-attention-based design instead. First, the 3D coordinates are transformed. For pixels with unknown depth, we learn a special C-dimensional embedding. For pixels with valid depth, their 3D points are linearly transformed to a C-dimensional vector. This results in a $16\times16\times C$ representation for each 16×16 patch. A transformer, shared across patches, converts each patch to a C-dimensional vector via a learned patch token which summarizes the patch [10]. This results in ${}^{W}/_{16} \times {}^{H}/_{16}$ tokens (and thus $N^{\text{enc}} = {}^{W}/_{16} \times {}^{H}/_{16} + 1$ with the additional global token used in a ViT [11]).

E^{RGB} **Patch Embeddings.** For RGB, we follow standard ViTs [11] and embed each 16×16 patch with a convolution.

Architecture. The E^{RGB} and E^{XYZ} encoder use a 12-layer 768-dimensional "ViT-Base" architecture [11, 65]. The input image size is 224×224. Our decoder is a lighter-weight 8-layer 512-dimensional transformer, following MAE [20]. Detailed specifications are in Supplementary Material.

4. Object Reconstruction Experiments

MCC works naturally for both objects and scenes. In §4, we show results and compare to competing methods for single object reconstruction. In §5, we show results on scenes.

Dataset. We use CO3D-v2 [51] as our main dataset for single object reconstruction. It consists of ~37k short videos of 51 object categories; the largest dataset of 3D objects in the wild. To show generalization to new objects, we hold out 10 randomly selected categories for evaluation and train on the remaining 41. The list of held-out categories is in Supplementary Material. Since CO3D is object-centric, we focus on foreground objects specified by segmentation masks provided in CO3D. Full 3D annotations, such as 3D meshes, are not available. CO3D extracts point clouds from the videos via COLMAP [55, 56], which are inevitably noisy and are used to train our model. Despite imperfect supervision, we show that MCC learns to reconstruct 3D shapes and texture and even corrects the noisy depth inputs.

Metrics. Following Kulkarni *et al*. [31], we report: accuracy (acc), the percentage of predicted points within ρ to a ground truth point, completeness (cmp), the percentage of ground truth points within ρ from a predicted point, and their F-score (F1) which drives our comparisons. ρ is 0.1.

Training Details. We train with Adam [29] for 150k iterations with an effective batch size of 512 using 32 GPUs, a base learning rate of 10^{-4} with a cosine schedule and a linear warm-up for the first 5% of iterations. Training takes ~2.5 days. We randomly scale augment images by $s \in [0.8, 1.2]$. We also perform 3D augmentations by randomly rotating 3D points along each axis by $\theta \in [-180^o, 180^o]$. Rotation is applied to the *seen* points P, the queries and the ground truth. Image I and points P are aligned through the concatenation of their encodings (Eq. 1). Points P and queries are consistent as well, as both are rotated. Essentially, our 3D augmentations build in rotation equivariance.

Coordinate System. We adopt the original CO3D coordinate system from [51], where objects are normalized to have zero-mean and unit-variance. Training and testing points are sampled from $[-3, 3]$ along each axis. Evaluation points are sampled with a granularity of 0.1.

4.1. Qualitative Results on Novel Categories

Fig. 4 shows qualitative results on the CO3D test set of novel categories. We show reconstructions for a variety of shapes and object types. MCC tackles heavy self-occlusions, *e.g.*, the mug handle is barely visible in the input image, and complex shapes, *e.g.*, the toy airplane. In addition to shape, MCC predicts texture which is difficult especially for unseen regions. For instance, the left and back side of the kids backpack is completely invisible, but MCC predicts to propagate the color from the right side. We also note that MCC is robust to noisy depth from COLMAP,

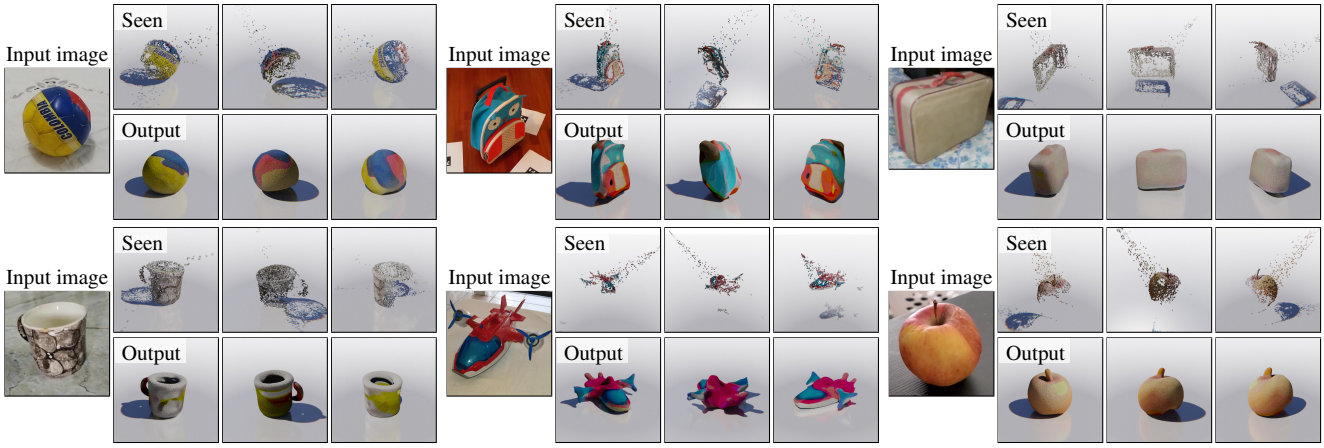


Figure 4. **Predictions on CO3D-v2 *Novel* Categories.** For each example, we show the input image (left), the unprojected seen 3D points (top), and our reconstruction (bottom). We show results for a variety of object types, shapes, textures and occlusion patterns. We emphasize that we do *not* use any shape priors such as symmetries, canonical views, or mean shapes. See project page for animations.

	Acc	Cmp	F1
Shared	42.6	77.0	52.5
Decoupled (ours)	47.5	76.0	**56.7**

(a) **Encoder Structure**

	Acc	Cmp	F1
MLP	43.4	79.8	54.5
PointNet [45]	45.6	80.3	56.6
Transformer (ours)	47.5	76.0	**56.7**

(b) E^{XYZ} **Design**

	Acc	Cmp	F1
Contrastive	45.0	78.7	55.6
Uniform (ours)	47.5	76.0	**56.7**

(c) **Training Query Sampling**

	Acc	Cmp	F1
Loc-pooled	49.2	22.6	28.2
Global	44.7	77.1	54.5
Detailed (ours)	47.5	76.0	**56.7**

(d) **Feature Conditioning**

	Acc	Cmp	F1
Loc+MLP	49.2	22.6	28.2
Cross-attn	42.3	49.5	43.7
Concat+attn (ours)	47.5	76.0	**56.7**

(e) **Decoder Design**

	Acc	Cmp	F1	CD (↓)
PoinTr [79]	79.6	27.1	39.7	0.065
MCC (w/o RGB)	46.5	70.8	53.9	0.047
MCC	47.5	76.0	**56.7**	**0.040**

(f) **Comparison to Prior Work with Explicit Design**

Table 1. **Ablations on CO3D-v2**, which validate MCC's design choices. We highlight ablation (e) which shows that an attention-based decoder outperforms an MLP, and (f) where we find that MCC's queriable decoder performs better than an explicit design [79]. Higher is better for Accuracy (Acc), Completeness (Cmp), and F1. Lower is better for Chamfer distance (CD).

present at varying degrees and depicted in the *seen* points of each example (top row). MCC corrects and completes the geometry in spite of the noise in depth inputs. We emphasize that we do not make geometric assumptions nor use any priors such as symmetry or mean templates when reconstructing objects. MCC learns only from data.

4.2. Ablation Study

Encoder Structure. In Table 1a, we ablate our encoder design which models I and P with two separate transformers (decoupled) and compare to a shared transformer which models the fused (sum) patch embeddings of I and P (shared). Our decoupled design performs slightly better.

E^{XYZ} Design. Table 1b compares our transformer to an MLP and PointNet [45] for the E^{XYZ} encoder. PointNet and our transformer, which model point interactions, work slightly better than an MLP, though not critically.

Training Query Sampling. In Table 1c, we compare our uniform sampling strategy with a contrastive-style sampling, where each example samples a fixed number of positives and negatives. Both work similarly. We choose uniform sampling because of its simplicity.

Feature Conditioning. Our input encoding R uses all N^{enc} tokens from the appearance I and geometry P encodings. We call this detailed conditioning and compare it with two popular choices: one where a globally average-pooled vector is used, as in [37, 42], and one where the feature vector is bilinearly interpolated at the projected location in the feature map, as in [21, 47, 78]. Table 1d validates our choice.

Decoder Design. As described in §3, MCC's decoder concatenates queries to the input encoding R in the token dimension, and a transformer models their interactions (concat+attn). We compare this design with two popular ones. Recent works on image-conditioned NeRF [21, 47, 78] condition points on their projected location in the feature map followed by an MLP (loc+MLP) – this comparison was also presented in the context of feature conditioning strategies. Another approach is cross-attention (cross-attn), where the encoded input R only serves as *keys/values* but not as *queries* to a transformer, *e.g.*, in Perceiver models [23, 24]. Table 1e shows that our decoder is critical for performance.

Comparison to Prior Work with an Explicit Design. Finally, we compare MCC and its queriable 3D decoder with a state-of-the-art 3D point completion method PoinTr [79]. PoinTr inputs an incomplete point cloud and predicts a

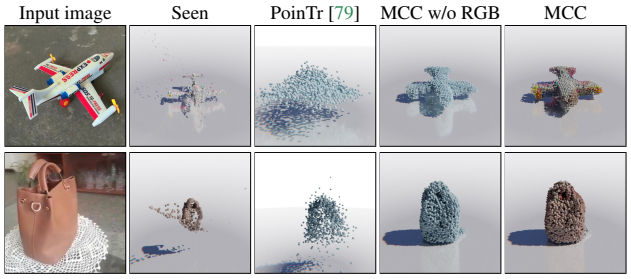
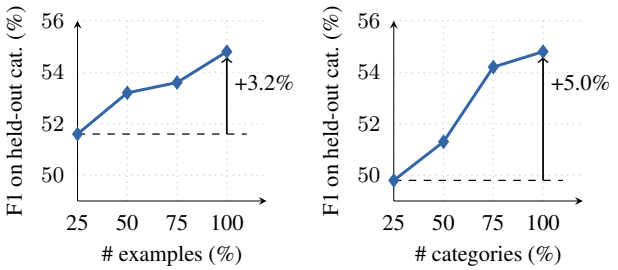


Figure 5. **Qualitative Comparison to PoinTr [79].** MCC predicts shape details while PoinTr tends to place points roughly around the object. For a fair comparison, MCC predicts the same number of points as PoinTr. Unlike PoinTr, MCC also predicts color.



(a) **Scaling # examples (same categories)** (b) **Scaling # categories**

Figure 6. **Scaling Behavior Analysis.** We train MCC on (a) a varying number of examples uniformly sampled from all training categories and (b) all examples from a varying number of training categories. All models are evaluated on the same held-out set of novel categories. We see clear performance gains from scaling training data, especially when expanding the number of categories. This supports that category-agnostic models and large-scale training are promising for 3D reconstruction.

fixed-resolution output using a transformer which models explicit geometric point relations (via nearest neighbors). We train PoinTr on CO3D which inputs the set of *seen* points P. For a fair comparison, we implement PoinTr with the same 12-layer architecture as ours, which is stronger than their 6-layer one. Since PoinTr does not use RGB, we compare with a MCC variant that ignores texture by encoding P but not I. We additionally report chamfer distance (CD), as in [79], and use the same number of points for a fair comparison. Table 1f shows that MCC outperforms PoinTr by a large margin. Fig. 5 presents a qualitative comparison. In §4.5, we also compare to NeRF-based methods.

4.3. Scaling Behavior Analysis

MCC's strength is that it only requires *points* for training and does not rely on any shape priors. As a result, MCC can train on a large number of examples. We analyze our model's performance as a function of data size. Fig. 6 shows that scaling the training data leads to steady performance improvements. Furthermore, if we increase the number of categories, and thus the visual diversity of our training data, the improvements are even larger. This suggests two things. First, building category-agnostic scaleable models like MCC is a promising direction towards general-purpose 3D reconstruction. Second, expanding the datasets, and especially the set of categories, is promising.

	depth sup. [9]	depth in	seen categ. Abs	seen categ. MSE	unseen categ. Abs	unseen categ. MSE
NeRF-WCE [21]			8.43	175.5	10.1	156.4
NeRF-WCE [21]	✓		7.38	92.2	9.15	139.9
NeRF-WCE [21]		✓	7.46	156.3	8.30	119.4
NeRF-WCE [21]	✓	✓	2.75	78.4	2.79	30.5
NerFormer [51]			2.02	70.4	2.00	20.6
NerFormer [51]	✓		2.19	72.8	2.18	23.5
NerFormer [51]		✓	2.20	72.1	2.17	22.5
NerFormer [51]	✓	✓	2.34	80.7	2.28	24.1
MCC	✓	✓	**1.46**	**38.8**	**1.17**	**13.6**

Table 2. **Comparison to the State-of-the-Art on CO3D-v2 [51].** For a fair comparison with MCC, we extend baselines [21,51] with depth supervision [9] or using depth as input. MCC outperforms prior state of the art on CO3D-v2 for shape reconstruction.

4.4. Zero-Shot Generalization In-the-Wild

In §4.1, we show generalization to novel categories from the CO3D dataset. Now, we turn to in-the-wild settings and show MCC reconstructions on ImageNet [8], iPhone captures, and AI-generated images [48].

iPhone Captures. This is arguably the most popular in-the-wild setting — our personal use of an off-the-shelf smart phone for capturing everyday objects. Specifically, we use iPhones and their depth sensor to take RGB-D images on a diverse set of objects in two of the coauthors' homes (using a 12 and 14 Pro iPhone). This is a challenging setting due to the domain shift from the training data and the difference in the depth estimation pipeline (COLMAP in CO3D *vs*. sensor from iPhone). Fig. 7a shows ours results. Examples such as the vacuum or the VR headset in Fig. 1b stand out as they deviate from our training set. Fig. 7a demonstrates MCC's ability to learn general shape priors, instead of memorizing the training set.

ImageNet. We turn to ImageNet [8], which contains highly diverse Internet photos, ranging from bears and elephants in their natural habitat to Japanese mailboxes, drastically different than the staged CO3D objects. For depth, we use an off-the-shelf model from Ranftl *et al*. [49], which differs from CO3D's COLMAP output. Fig. 7b shows results on ImageNet images of diverse objects.

AI-generated Images. We test MCC on DALL·E 2 which generates images of imaginary objects. Fig. 7c shows MCC reconstructions including the Internet-famous avocado chair and a cat-shaped marshmallow with a mustache!

4.5. Comparison to Image-Conditioned NeRF

A recent successful line of work for 3D reconstruction extends NeRF [38] to cross-scene generalization from one or few views by conditioning on image embeddings [21,47,

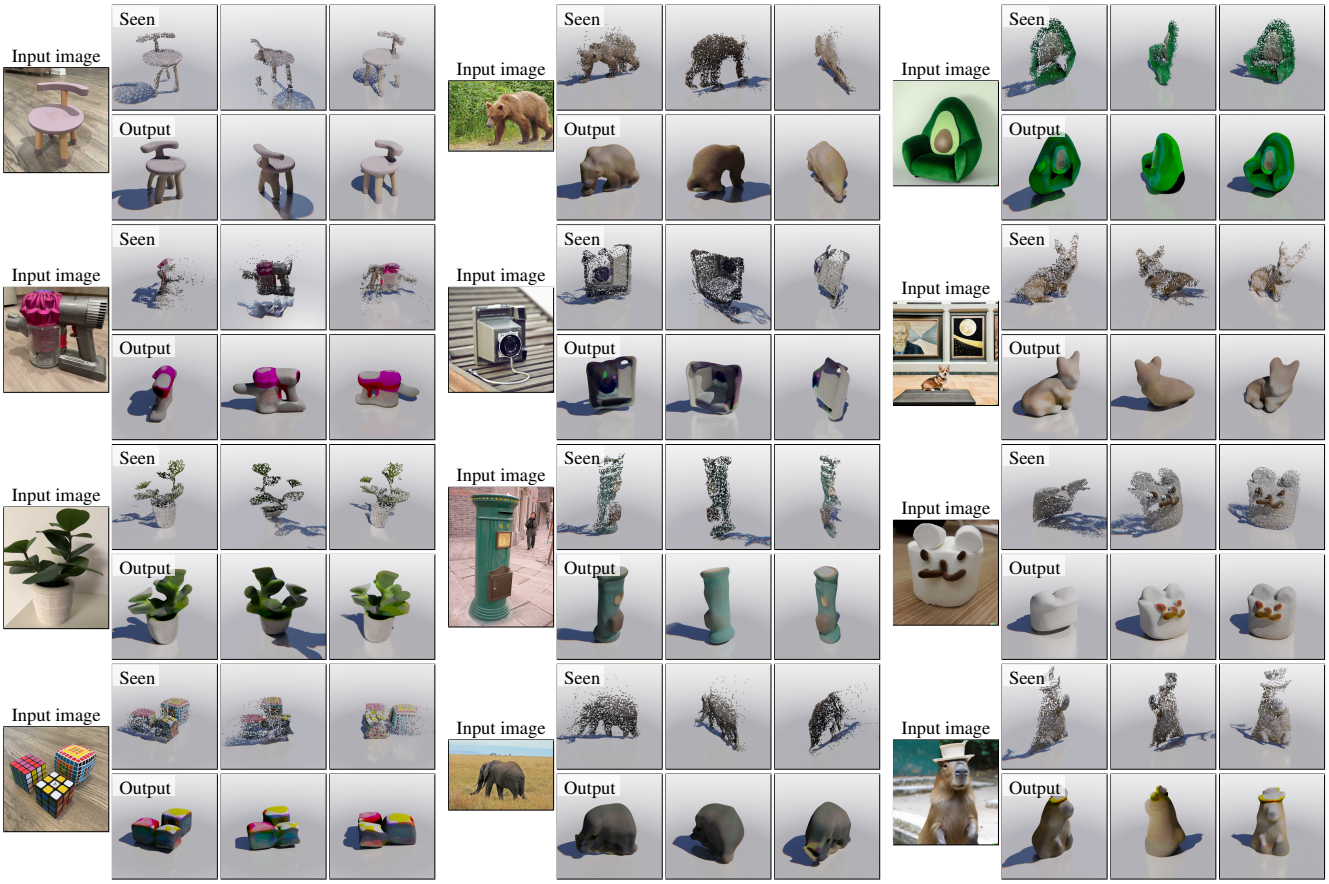


Figure 7. **Zero-Shot Generalization.** We test MCC, trained on CO3D-v2 [51], on three challenging settings: (a) iPhone captures with LiDAR sensor of everyday objects, (b) Web images (from ImageNet) of in-the-wild objects with depth estimated by an off-the-shelf model [49], (c) AI-generated images (by DALL·E 2) of imaginary objects with depth estimated by [49]. These examples are challenging as they demonstrate variance in object types (*e.g.*, novel, imaginary objects), image styles (*e.g.*, digital arts, natural), depth systems (*e.g.*, depth sensor, off-the-shelf predictors), and visual context (*e.g.*, safari, street scene). See project page for animations.

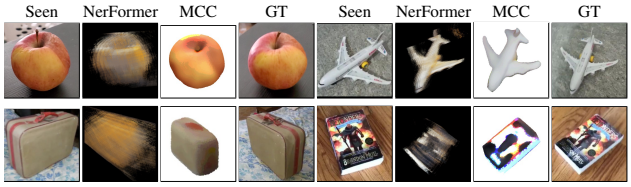


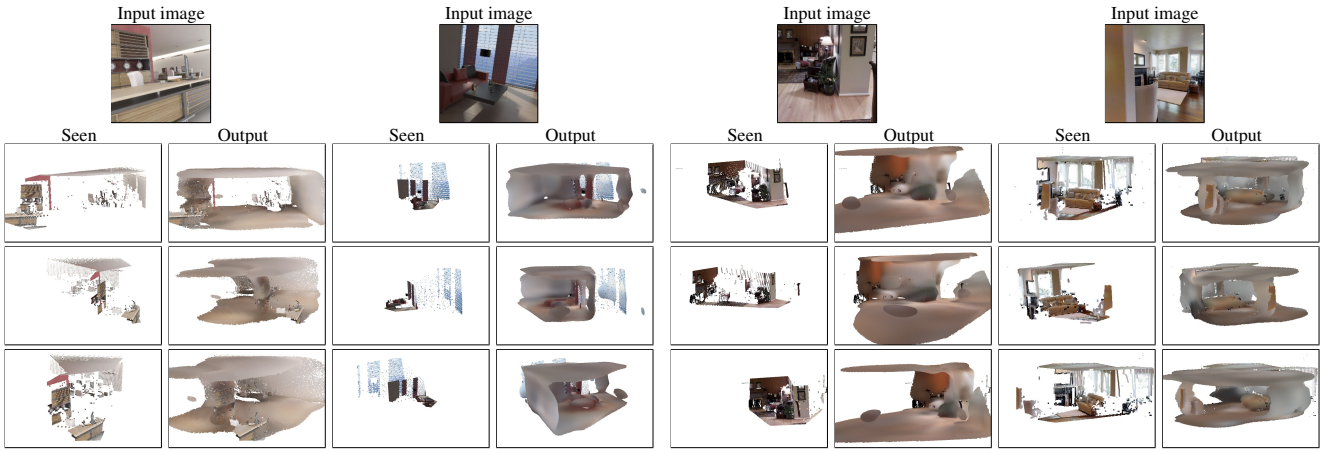
Figure 8. **Qualitative Comparison between MCC and NerFormer [51].** NerFormer captures texture but struggles with geometry; MCC reconstructs shapes more accurately.

51, 78]. We compare to two recent best performing methods on CO3D from this family, NeRF-WCE [21] and NerFormer [51]. We evaluate for shape reconstruction using the official CO3D novel view depth metrics [51]: absolute (abs) and mean-squared error (MSE) on the official CO3D challenge evaluation frames. This puts MCC at a disadvantage as it is not designed for synthesis via rendering. Since MCC uses RGB-D as input, we extend both methods, which originally use posed RGB views, to take depth as input or supervision. For depth supervision we follow Deng *et al*. [9], which shows strong results by supervising NeRF models with depth. To input depth, we fuse the XYZ input encoding, *i.e.* $E^{\text{XYZ}}(P)$, to the input image features. Table 2 shows that the baselines benefit from depth, as expected; MCC outperforms them by a clear margin. Fig. 8 qualitatively compares MCC to the best baseline, NerFormer [51]. NerFormer captures texture but struggles with geometry under the challenging single-view novel-category setting, thus rendering relatively blurry novel views. Admittedly, these methods tend to work better with more (5-10) input views. MCC predicts more accurate shapes from just a single view.

5. Scene Reconstruction Experiments

MCC naturally handles singles objects and scenes without modifications to its design. So, now we turn to scenes.

Task. We test 3D scene reconstruction from a single RGB-D image. Formally, we aim to reconstruct everything in front of the camera ($z > 0$ in camera coordinate system) up to a certain range. Note that this includes areas outside the camera frustum, which increases the complexity of the task.



(a) Reconstruction of Hypersim Held-Out Scenes (b) *Zero-Shot* Generalization to the Taskonomy Dataset

Figure 9. **Scene Reconstructions.** With a model trained on Hypersim, we show reconstructions on (a) held-out Hypersim scenes, and (b) novel scenes from Taskonomy. From a single RGB-D image, MCC reconstructs furniture, walls, floors, and ceilings, even outside the view frustum. Capturing fine scene details is hard, but more data can help as our analysis in §4.3 suggests. See project page for animations.

	Acc	Cmp	F1
DRDF [31]	54.4	1.0	2.0
DRDF (our arch) [31]	54.2	1.4	2.7
MCC	**66.3**	**1.5**	**2.8**

Table 3. **Comparison to DRDF on Hypersim.** MCC outperforms the state-of-the-art scene reconstruction approach, DRDF [31], extended to input depth, with both its original and our architecture.

Dataset. We experiment on the Hypersim dataset [52], which contains complex, diverse scenes, such as warehouse, lofts, restaurants, church *etc*., with over 77k images. We split the dataset into 365 scenes for training and 46 scenes for testing. We use images along with the associated depth as ground truth for training. Since 3D meshes are available, we use them for evaluation and report the metrics from §4.

5.1. Hypersim Scene Reconstruction

Qualitative Results. Fig. 9a shows qualitative results on Hypersim [52]. While MCC only sees the scene within the view frustum, it is able to complete furniture, walls, floors, and ceilings. For instance, in the left example, MCC predicts the space behind the kitchen, including the floors, which are almost entirely occluded in the input view. In the right example, MCC predicts the wall on the left which is entirely outside of the view frustum. Scene reconstruction from a single view is hard; while MCC reconstructs the room geometry it fails to capture fine details in both shape and texture. We expect more data to significantly improve performance, as suggested by our scaling analysis in §4.3.

Quantitative Evaluation. We compare to recent state-of-the-art on scene reconstruction, DRDF [31], which we extend to take RGB-D inputs like MCC. Table 3 shows that MCC outperforms DRDF across all metrics. We also extend DRDF to use MCC's architecture but keeping its original loss and ray-based inference. This variant performs better than the original DRDF but still worse than MCC.

5.2. Zero-Shot Generalization to Taskonomy

Finally, we deploy MCC, trained on Hypersim, on novel scenes from Taskonomy [81]. While photorealistic, Hypersim is synthetic, while Taskonomy is real. So, we test both generalization to novel scenes but also the "sim-to-real" transfer. Fig. 9b shows MCC's reconstructions, which demonstrate that our model is able to reconstruct the room layout (floors, walls, ceilings) in this challenging setting.

6. Failure Cases

While MCC has demonstrated promising results, we observe three error modes: (1) Sensitivity to depth input. MCC can recover from noisy depth inputs. But if depth is largely incorrect, it will fail to reconstruct accurate 3D geometry. (2) Distribution shifts. For targets far from the training distribution, we see errors in texture and geometry (*e.g*., Rubik's cubes). (3) High-fidelity texture. Detailed texture predictions from a single view are difficult and MCC often omits details (*e.g*., text on volleyball in Fig. 4).

7. Conclusions

We present MCC, a general-purpose 3D reconstruction model that works for both objects and scenes. We show generalization to challenging settings, including in-the-wild captures and AI-generated images of imagined objects. Our results show that a simple point-based method coupled with category-agnostic large-scale training is effective. We hope this is a step towards building a general vision system for 3D understanding. Models and code are available online.

From an ethics standpoint, as with all data-driven methods, MCC can potentially inherit the bias (if any) in data. In this project, we solely train on inanimate objects and scenes to minimize the risk. We do not foresee immediate negative repercussions with the model, but caution against future use without paying careful attention to the training dataset.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: Bert pre-training of image transformers. In *ICLR*, 2022. 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 1, 2

[4] Anh-Quan Cao and Raoul de Charette. MonoScene: Monocular 3D semantic scene completion. In *CVPR*, 2022. 2

[5] Jose A Castellanos, José MM Montiel, José Neira, and Juan D Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on robotics and Automation*, 1999. 2

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2

[7] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3D objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2, 6

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 2, 6, 7

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2, 4

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3, 4

[12] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 2004. 2

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017. 2

[14] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, 2016. 2

[15] Andreas Geiger and Chaohui Wang. Joint 3D object and layout inference from a single RGB-D image. In *German Conference on Pattern Recognition*, 2015. 2

[16] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[17] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *ICCV*, 2019. 1, 2

[18] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoints without keypoints. In *ECCV*, 2020. 2

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 3, 4

[21] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3D object categories from videos in the wild. In *CVPR*, 2021. 2, 3, 5, 6, 7

[22] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point fractal network for 3D point cloud completion. In *CVPR*, 2020. 2

[23] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 5

[24] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 5

[25] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH*, 1984. 2

[26] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 1, 2

[27] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, 2018. 2

[28] Young Min Kim, Niloy J Mitra, Dong-Ming Yan, and Leonidas Guibas. Acquiring 3D indoor environments with variability and repetition. *TOG*, 2012. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[30] Nilesh Kulkarni, Abhinav Gupta, David Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, 2020. 1, 2

[31] Nilesh Kulkarni, Justin Johnson, and David F. Fouhey. What's behind the couch? directed ray distance functions for 3D scene reconstruction. In *ECCV*, 2022. 2, 4, 8

[32] Abhijit Kundu, Yin Li, and James M. Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *CVPR*, 2018. 1

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[34] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[35] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. In *ICCV*, 2019. 2

[36] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 1982. 4

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019. 2, 3, 5

[38] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 6

[39] Aron Monszpart, Nicolas Mellado, Gabriel J Brostow, and Niloy J Mitra. RAPter: rebuilding man-made scenes with regular arrangements of planes. *TOG*, 2015. 2

[40] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *TOG*, 2012. 2

[41] Ramakant Nevatia and Thomas O Binford. Description and recognition of curved objects. *Artificial intelligence*, 1977. 1

[42] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *CVPR*, 2020. 3, 5

[43] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *ICCV*, 2019. 1, 2

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2, 3

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 5

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[47] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *CVPR*, 2021. 2, 3, 5, 6

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2, 6

[49] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 6, 7

[50] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arXiv:2007.08501*, 2020. 2

[51] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021. 2, 4, 6, 7

[52] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 2, 8

[53] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 2, 3

[54] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002. 2

[55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 4

[56] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2, 4

[57] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. DeepVoxels: Learning persistent 3D feature embeddings. In *CVPR*, 2019. 2

[58] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, 1990. 2

[59] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2

[60] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 1, 2

[61] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 1, 2

[62] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992. 2

[63] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 2008. 2

[64] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with PixelCNN decoders. *NeurIPS*, 2016. 3

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 3, 4

[66] Bram Wallace and Bharath Hariharan. Few-shot generalization for single-image 3D reconstruction via priors. In *ICCV*, 2019. 2

[67] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z Jane Wang, and Rabab Ward. Multi-view 3d reconstruction with transformers. In *ICCV*, 2021. 2

[68] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, 2018. 1, 2

[69] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In *ICCV*, 2019. 2

[70] Charles Wheatstone. Contributions to the physiology of vision.—part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical transactions of the Royal Society of London*, 1838. 2

[71] Markus Worchel, Rodrigo Diaz, Weiwen Hu, Oliver Schreer, Ingo Feldmann, and Peter Eisert. Multi-view mesh reconstruction with neural deferred shading. In *CVPR*, 2022. 2

[72] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. MarrNet: 3D shape reconstruction via 2.5D sketches. *NeurIPS*, 2017. 2

[73] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020. 1

[74] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3D reconstruction from single and multi-view images. In *ICCV*, 2019. 2

[75] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. *NeurIPS*, 2019. 2

[76] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. ShapeFormer: Transformer-based shape completion via sparse representation. In *CVPR*, 2022. 2

[77] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. *NeurIPS*, 2016. 2

[78] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 3, 5, 6

[79] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 2, 5, 6

[80] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *3DV*, 2018. 2

[81] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2, 8

[82] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[83] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1

[84] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2