Omnidata: A Scalable Pipeline for Making Multi-Task Mid-Level Vision Datasets from 3D Scans

Ainaz Eftekhar[†*] Alexander Sax[‡*] Jitendra Malik[‡] Amir Zamir[†]

[†]Swiss Federal Institute of Technology (EPFL) [‡]University of California, Berkeley

https://omnidata.vision

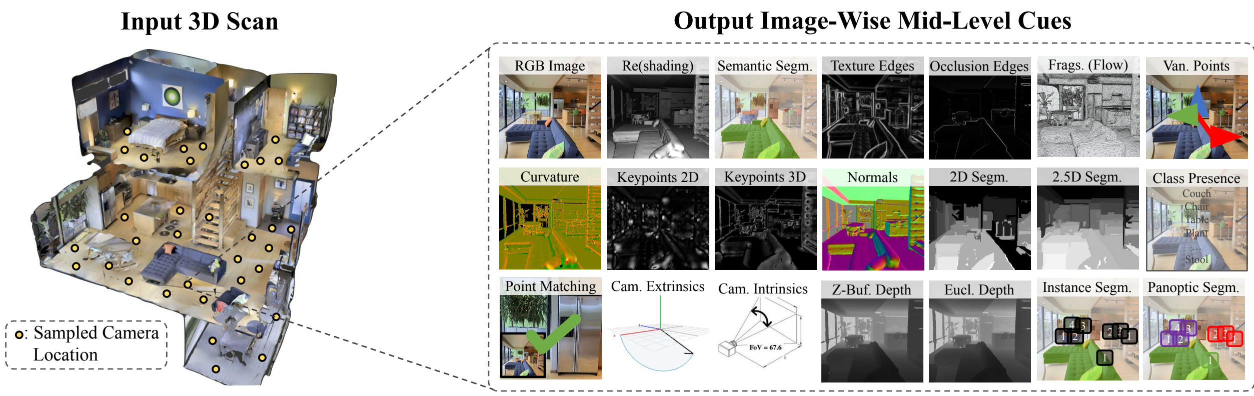

Figure 1: **Generating mid-level cues from real-world 3D scans**. **(Left:)** The proposed pipeline generates dense camera positions and points-of-interest in the input space and **(right)** renders 21 image-wise mid-level cues by default. Models trained on a starter dataset of this programmatically generated data roughly matches human performance for surface normal estimation on OASIS [12] (see Sec. 4).

Abstract

This paper introduces a pipeline to parametrically sample and render static multi-task vision datasets from comprehensive 3D scans from the real-world. In addition to enabling interesting lines of research, we show the tooling and generated data suffice to train robust vision models. Familiar architectures trained on a generated starter dataset reached state-of-the-art performance on multiple common vision tasks and benchmarks, despite having seen no benchmark or non-pipeline data. The depth estimation network outperforms MiDaS and the surface normal estimation network is the first to achieve human-level performance for in-the-wild surface normal estimation—at least according to one metric on the OASIS benchmark.

The Dockerized pipeline with CLI, the (mostly python) code, PyTorch dataloaders for the generated data, the generated starter dataset, download scripts and other utilities are all available through our project website.

*Equal contribution.

1. Introduction

This paper introduces a pipeline to bridge the gap between comprehensive 3D scans and static vision datasets. Specifically, we implement and provide a platform that takes as input one of the following:

- a textured mesh,
- a mesh with images from an actual camera/sensor,
- a 3D pointcloud and aligned RGB images,

and generates a multi-task dataset with as many cameras and images as desired to densely cover the space. For each image, there are 21 different default mid-level cues, shown in Fig. 1. The software makes use of Blender [16], a powerful physics-based 3D rendering engine to create the labels, and exposes complete control over the sampling and generation process. With the proliferation of reasonably-priced 3D sensors (e.g. Kinect, Matterport, and the newest iPhone), we anticipate an increase in such 3D-annotated data.

In order to establish the soundness for training computer vision models, we used our pipeline to annotate several existing 3D scans and produce a medium-size starter dataset of mid-level cues. Samples of the data and different cues are shown in Fig. 5. Standard models trained on this starter dataset achieve state-of-the-art performance for several standard computer vision tasks. For surface normal estimation, a standard UNet [45] model trained on this starter dataset yields human-level surface normal estimation performance on the in-the-wild dataset OASIS [12], even though the model never saw OASIS data during training. For depth estimation, our DPT-Hybrid [41] is comparable to or outperforms state-of-the-art models such as MiDaS DPT-Hybrid [42, 41]. The qualitative performance of these networks (shown in Figs. 6, 7) is often better than the numbers suggest, especially for fine-grained details.

We further provide an ecosystem of tools and documentation around this platform. Our project website contains links to a Docker containing the annotator and all necessary libraries, PyTorch [39] dataloaders to efficiently load the generated data, pretrained models, scripts to generate videos in addition to images, and other utilities.

We argue that these results should not be interpreted narrowly. The core idea of the platform is that the "sectors of the ambient [light-field] array are not to be confused with temporary *samples* of the array" (J. J. Gibson [21]). That is, static images only represent single samples of the entire 360-degree panoramic light-field environment surrounding an agent. How an agent or model samples and represents this environment will affect its performance on downstream tasks. The proposed platform in this paper is designed to reduce the technological barriers for research into the effect of data sampling practices and into the interrelationships between data distribution, data representation, models, and training algorithms. We discuss directions here and analyze a few illustrative examples in the final section of the paper.

First, the pipeline proposed in this paper provides a possible pathway to understand such sampling effects. That is, the rendering pipeline offers complete control over (heretofore) fixed design choices such as camera intrinsics, scene lighting, object-centeredness [40], the level of "photographer's bias" [6], data domain, and so on. This makes it possible to run intervention studies (e.g. A/B tests), without collecting and validating a new dataset or relying on a post-hoc analysis. As a consequence, this provides an avenue for a computer vision "dataset design guide".

Second, vision is about much more than semantic recognition, but our datasets are biased towards that as the core problem. The best-studied, most diverse and largest dataset (>10M images) generally contains some form of textual/class labels [18, 51] and only RGB images. On the other hand, datasets for most non-classification tasks remain tiny by modern standards. For example, the indoor scene dataset NYU [47], still used for training some state-of-the-art depth estimation models [62], contains only 795 training images—all taken with a single camera. The pipeline presents a way to generate datasets of comparable quality for non-recognition tasks.

Third, the generated data allows "matched-pair experimental design" that simplifies study into the *interrelationships* of different tasks, since the pipeline produces labels for every sample. In particular, it helps to avoid issues like the following: suppose a model trained for object classification on ImageNet transfers to COCO [32] better than a model trained for depth estimation on NYU [47]–is that due to the data domain, the training task, the diversity of camera intrinsics, or something else?

Existing matched-pair datasets usually focus on a single domain (indoor scenes [64, 47, 3, 50], driving [20, 17], block-worlds [24], etc.) and contain few cues [17, 47, 3, 50]. The provided starter dataset may be a better candidate for this research than these existing datasets, since it contains over 14.5 million images from different domains (more than the full ImageNet database), contains many different cues (e.g. for depth, surface normals, curvature, panoptic segmentation, and so on), and models trained on this dataset reach excellent performance for several tasks and existing benchmarks. We demonstrate the value of such matched-pairs data in Sec. 5.3,

Though our pipeline is designed to facilitate understanding the principles of dataset design, vision beyond recognition, the interrelationships between data, tasks, and models, this paper does not extensively pursue those questions themselves. It provides a few analyses, but these are merely intended as illustrative examples. Instead, the paper introduces tooling designed to facilitate such research as 3D data becomes more widely available and the capture technology improves. On our website, we provide a documented, open-sourced, and Dockerized annotator pipeline with a convenient CLI, runnable examples, the starter dataset, pretrained models, PyTorch dataloaders, and code for the paper (including annotator and models).

2. Related Work

In this section we examine related datasets and other approaches. A thorough review would take more space than we have, so we restrict our attention to only the most relevant groupings.

Static 3D Datasets. The past few years have witnessed an uptick in the number of mesh-based datasets, thanks largely to the availability of reasonably-priced 3D scanners. Each dataset in the current crop, though, usually consists of scenes in a restricted domain. Prominent examples of indoor building datasets include Stanford Building Dataset (S3DIS) [5], Matterport3D [9], Taskonomy [64], Replica [50], 2D-3D-Semantic [4], Habitat-Matterport [36], and Hypersim [44]. Other datasets contain primarily outdoor scenes, usually driving–such as CARLA [20], GTA5 [43]–or other narrow domains such as the aptly-named Tanks and Temples [28] dataset. Models trained on such scene-level views often do not generalize

Cue Generation Pipeline

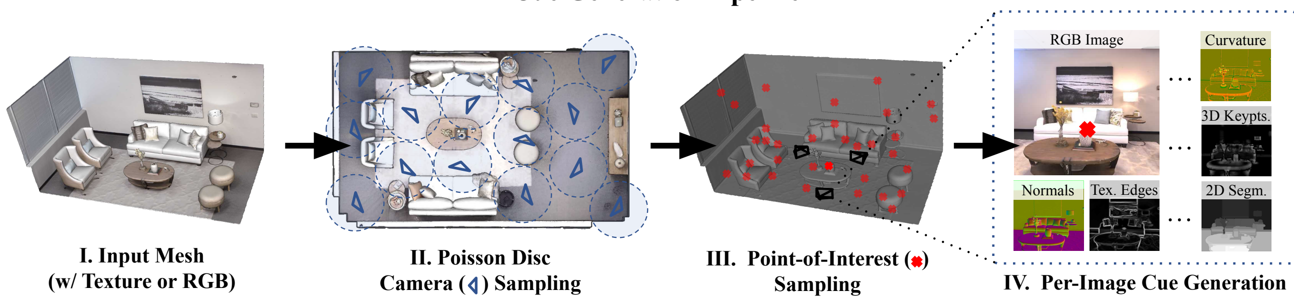

Figure 2: **Overview of the generation pipeline.** (**I**) Given a textured mesh (or other options discussed in Sec. 3.1), our pipeline (**II**) generates dense camera locations, (**III**) generates points-of-interest subject to multi-view constraints and (**IV**) produces 21 different mid-level cues for each (shown in Fig. 1).

to object-centric views (see Fig. 7), but existing datasets with high-resolution object meshes do not include 2D images samples [1, 8].

Other recent datasets aim to link diverse monocular 2D images and corresponding 3D meshes, but take the reverse approach of this paper by using hand-annotation to create meshes from single-view in-the-wild RGB samples [12, 13]. This labeling process is expensive and time-consuming, and crucially does not allow regenerating the image dataset. In Sec. 4.3, we consider our pipeline vs. OASIS, one of the largest and most diverse of these benchmarks, and demonstrate that models trained on our starter dataset already match human-level performance on OASIS—outperforming the same architectures models trained on OASIS itself.

Vision-Focused Simulators. Like our platform, simulators typically take a textured mesh as the representation of the scene and aim to produce realistic sensory inputs [36, 59]. While spiritually similar to the pipeline proposed in this paper, the current generation of simulators is designed first and foremost to train embodied agents. They prioritize rendering speed and real-time mechanics at the cost of photorealism and cue diversity [25, 38]. Extending such simulators to handle additional cues or to parametrically render out vision datasets often requires writing new components of the simulator codebase (usually in C++, CUDA, or OpenGL), a surmountable but unpleasant barrier to entry. In contrast, our platform extends Blender which "supports the entirety of the 3D pipeline" [15] and provides Python bindings that will be intuitive to most vision researchers, and we implement many of these cues and sampling methods out-of-the-box. In short, we provide a bridge between simulators and static vision datasets.

Multi-Task Datasets. Vision-based multi-task learning (MTL), like computer vision in general, shows a general bias towards recognition. MTL datasets often take different shades of classification as the core problem of interest [30, 57, 35]. In particular, MTL literature often focuses on binary attribute classification in specialized domains, such as Caltech-UCSD Birds [58] or CelebA [34]. Visual MTL datasets that contain non-recognition tasks often contain only a single domain or a few tasks (NYU [47], CityScapes [17] or Taskonomy [64]). Sometimes, MTL papers take mix datasets for a "single" task and consider each dataset as a different task [33, 42, 31, 41].

In general, the multi-task learning literature has not converged on a standardized definition of the setting or dataset. Recent work has demonstrated that MTL methods developed on existing datasets seem to specialize to their respective developement set and do not perform well on large, realistic datasets, or on other tasks [55, 56, 65]. This underscores the importance of developing *realistic training setting and datasets* that generalizes to real-world scenarios.

Data Augmentation + Domain Randomization. Data augmentation is a way to modify the data or training regimen so that the trained model exhibits desirable invariances (or equivariances). Specifically, *any* transformation of sensor inputs that determines a corresponding (possibly identity) transformation on the label can be used as "augmented" data. Simple 2D augmentations such as 2D affine transformations, crops, and color changes are by far the most common in computer vision [10, 22], since they can be used even when datasets lack 3D geometry information. In robotics and reinforcement learning where 3D simulators are more standard, data augmentation was introduced as "domain randomization" [53], and common augmentations include texture and background randomization on the scene mesh. Recently, [19] introduced a Blender-based approach for doing domain randomization and creating static datasets of RGB, depth, and surface normals from SunCG [52].

Our pipeline makes all these augmentations available for static computer vision datasets: not just flips/crops/texture randomization, but also dense viewpoints, multi-view consistency, Euclidean transforms, lens flare, etc.). We implement and examine depth-of-field augmentation in Sec. 5.1.

3. Pipeline Overview

We call our pipeline `Omnidata` as it aspires to encapsulate comprehensive scene information (“omni”) in the generated “data”. **Try a live example here** to get acquainted with the pipeline. The example uses the CLI and a YAML-like config file to generate images from a textured mesh in Replica [50].

Inputs: The annotator operates upon the following inputs:

- Untextured Mesh (.obj or .ply)
- *Either:* Mesh Texture or Aligned RGB Images
- *Optional:* Pre-Generated Camera Pose File

A 3D pointcloud can be used as well: simply mesh the pointcloud using a standard mesher like COLMAP [46]. An example of meshing and using a 3D pointcloud with the annotator, as well as a more complete description of inputs are available in the supplementary.

Outputs: The pipeline generates 21 mid-level cues in the initial release. All labels are available for all generated images (or videos). Fig. 1 provides a visual summary of the different types of outputs. A detailed description of the default mid-level cues and additional outputs provided by the `Omnidata` annotator is included in the supplementary.

3.1. Sampling and Generation

In this section we provide a high-level outline of the generation and rendering process (see Fig. 2), deferring full details to the supplementary.

First, the annotator generates camera locations (Fig. 2 II) and points-of-interest (Fig. 2 III) along the mesh.

Second, for each camera and each point-of-interest, it creates a view from that camera fixated on the point (three fixated views are depicted in the lower part of Fig. 3).

Third, for each space-point-view triplet, the annotator renders (Fig. 4) all the mid-level cues (Fig. 2 IV).

Each step is elaborated next.

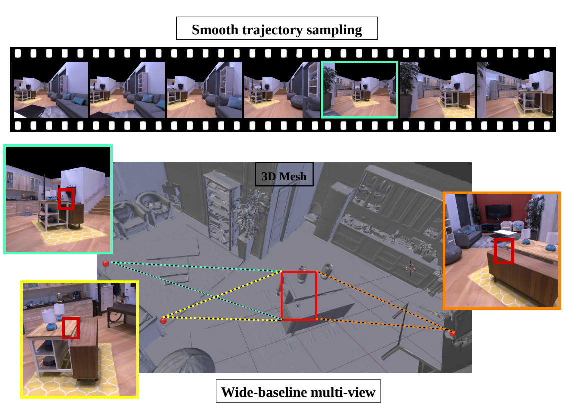

Figure 3: **Wide- and narrow-baseline dense view sampling.** Each point-of-interest can be viewed by a guaranteed minimum number of cameras. We also provide an option for creating denser views with narrower baselines (*e.g.* similar to consecutive video frames) that is crucial for inverse rendering methods..

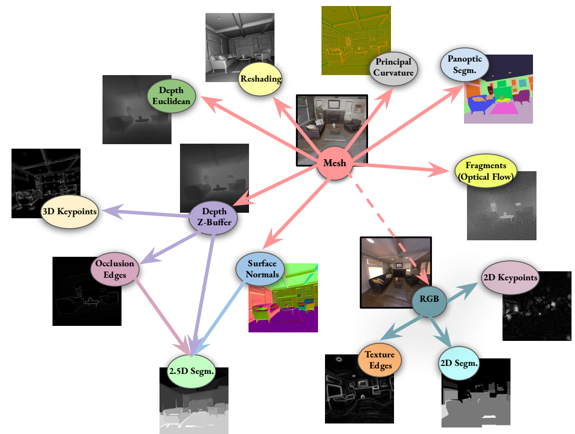

Figure 4: **DAG of processing pipeline.** The pipeline uses some of the mid-level cues to produce others. The ordering of this process (for image-like cues) is shown by the DAG.

Camera and Point Sampling: Camera locations can be provided (if the mesh comes with aligned RGB), or as in Fig. 2 II, the annotator generates cameras in each space so that cameras are not inside or overlapping with the mesh (default: cameras generated via Poisson-disc sampling to cover the space). Points-of-interest are then sampled from the mesh with a user-specified sampling strategy (default: uniform sampling of each mesh face and then uniform sampling on that face). Cameras and points are then filtered so that each camera sees at least one point and each point is seen by at least some user-specified minimum number of cameras (default: 3).

View Sampling: The annotator provides two default methods for generating views of each point. The first method (wide-baseline) generates images while the second, (smooth-trajectory mode) generates videos.

- *Wide-baseline multi-view:* A view is saved for each space-camera-point combination where there is an unobstructed line-of-sight between the camera center and the point-of-interest. The camera is fixated on the point-of-interest, as shown in Fig. 3, bottom.
- *Smooth trajectory sampling:* For each point of interest, a subset of cameras with a fixated view of the point are selected, and a smooth cubic-spline trajectory is interpolated between the cameras. Views of the point are generated for cameras at regular intervals along this trajectory (see Fig. 3, top).

Rendering mid-level cues: Since no single piece of software was able to provide all mid-level cues, we created an interconnected pipeline connecting several different pieces of software that are all freely available and open-source. We tried to primarily use Blender (a 3D creation suite), since it has an active user and maintenance community, excellent documentation, and python bindings for almost everything. Used by professional animators and artists, it is is generally well-optimized. The overall pipeline is fairly complex, so

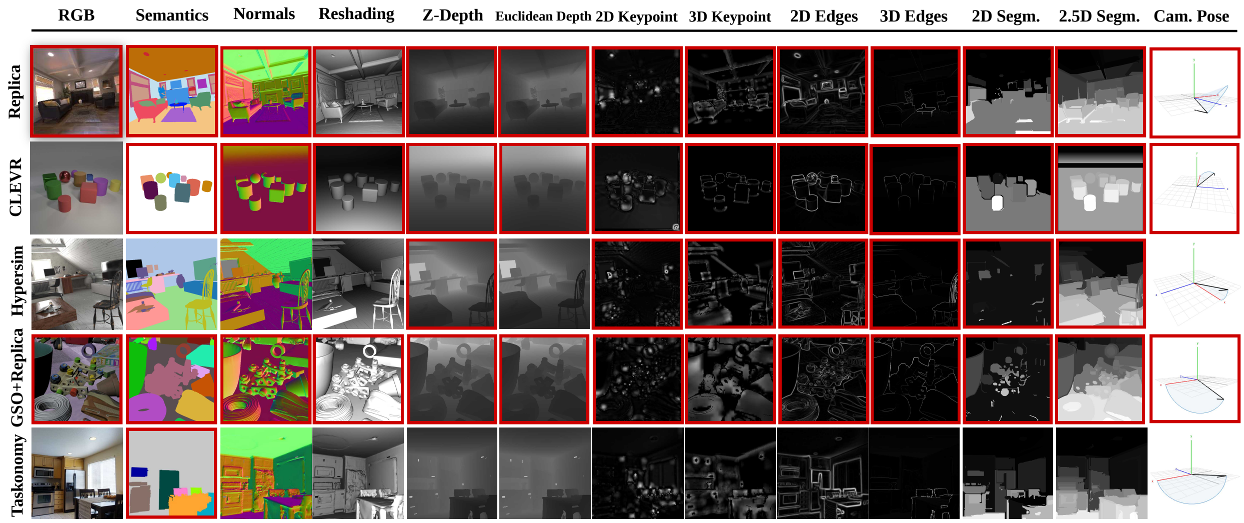

Figure 5: **Mid-level cues provided for the starter set.** 12 out of 21 mid-level cues visualized for each component dataset in the starter set, which contains both scenes and objects. Images with red borders indicate cues that were not included in the original data. Fig. 1 visualizes all 21 cues.

we defer a full description to the supplementary. The order of cue generation is shown in Fig. 4. The full code is available on our website.

Performance: The annotator generates labels in any resolution. Each space+point+view+cue label in the starter dataset (512×512) typically takes 1-4 seconds on server or desktop CPUs and can be parallelized over many machines.

3.2. Ecosystem Tools

To simplify adoption, the following tools are available on our website and the associated GitHub repository:

Pipeline code and documentation.

Docker containing the annotator and properly linked software (Blender [15], compatible Python versions, MeshLab [14], etc.).

Dataloaders in PyTorch for correctly and efficiently loading the resulting dataset

Starter dataset of 14.5 million images with associated labels for each task

Convenience utilities for downloading and manipulating data and *automatically filtering* misaligned meshes (description and sensitivity analysis in the supplementary).

Pretrained models and code, including the first publicly available implementation of MiDaS [42] training code.

4. Starter Dataset Overview

We provide a relatively large starter dataset of data annotated with the `Omnidata` annotator. The dataset comprises roughly **14.5 million** images from scenes that are both scene- and object-centric. Fig. 5 shows sample images from the starter dataset along with 12 of the 21 mid-level cues provided. Cues that are not present in the original dataset are indicated with a red border. We evaluate this starter dataset on existing benchmarks in Sec. 4.3. Note that the dataset could be straightforwardly extended to other existing outdoor and driving datasets such as GTA5 [43], CARLA [20], or Tanks and Temples [29].

4.1. Datasets Included

The starter data was created from 7 mesh-based datasets:

Indoor scene datasets: Replica [50], HyperSim [44], Taskonomy [64], Habitat-Matterport (HM3D)

Aerial/outdoor datasets: BlendedMVG [61]

Diagnostic/Structured datasets: CLEVR [24]

Object-centric datsets: To provide object-centric views in addition to scene-centric ones, we create a dataset of Google Scanned Objects [1] scattered around buildings from the Replica [50] dataset (similar to how ObjectNet [7] diversified images for image classification). We used the Habitat [36] environment to generate physically plausible scenes, and generated different densities of objects. Examples of images are shown in Fig. 5, and a full description of the generation process is available in the supplementary.

Dataset	Images Train	Images Val	Images Test	Spaces Train	Spaces Val	Spaces Test	Points
CLEVR	60,000	6,000	6,000	1	0	0	72,000
Replica	56,783	23,725	23,889	10	4	4	4,150
Replica + GSO	107,404	43,450	42,665	10	4	4	31,167
Hypersim	59,543	7,386	7,690	365	46	46	74,619
Taskonomy	3,416,314	538,567	629,581	379	75	79	684,052
BlendedMVG	79,023	16,787	16,766	341	74	73	112,576
Habitat-Matterport	8,470,855	1,061,021	-	800	100	-	564,328
Total (no CLEVR)	12,189,922	1,690,936	720,591	1,905	303	206	1,434,892

Table 1: **Component dataset statistics.** Breakdown of train/val/test split sizes in each of the components of the starter datset.

4.2. Dataset Statistics

The starter dataset contains **14,601,449** images from **2,414** spaces. Views are both scene- and object-centric, and they are labeled with each modality listed in Fig.1. Camera field-of-view is sampled from a truncated normal distribution between 30° and 125° with mean 77.5°, and camera

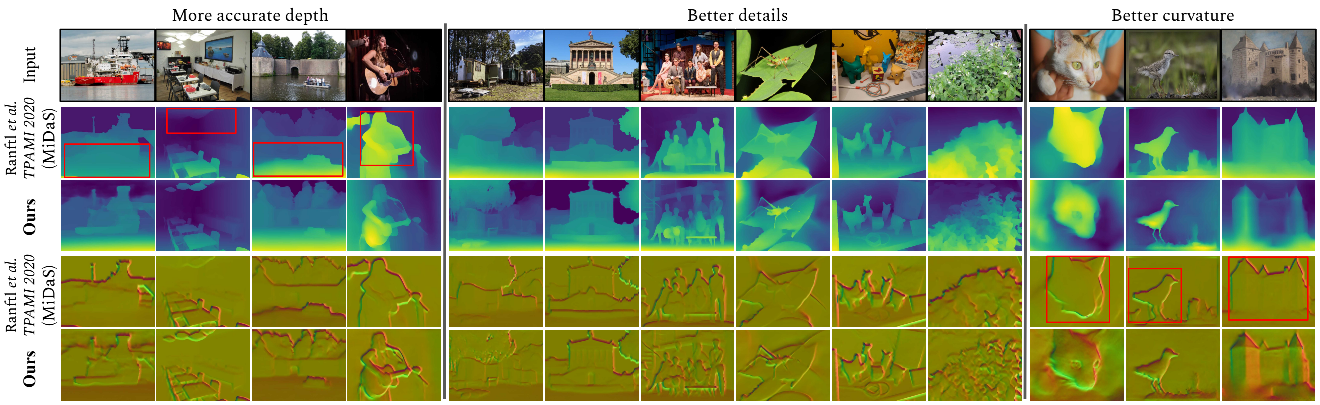

Figure 6: **Qualitative comparison with MiDaS on zero-shot OASIS depth estimation.** The last 2 rows show the surface normals extracted from the depth predictions. Our model predicts more accurate depth (left), and also outperforms in recovering the fine-grained details (middle). As shown by the extracted surface normal in the last 3 columns (right), our depth predictions better reflect the curvature and true shape of the objects, while the same regions appear flat in the predictions by the MiDaS model. The red rectangles highlight the regions useful for comparison [best viewed zoomed in].

roll is uniform in $[-10°, 10°]$. Tab. 1 contains data broken down to sub-datasets.

4.3. Soundness for Existing Computer Vision

We demonstrate that the generated dataset is capable of training standard, modern vision systems to state-of-the-art performance on existing benchmarks. Once we have established that the models can be trusted, we further provide a few transfer experiments to quantify how related the different component datasets are.

We show that the models trained on a 5-dataset portion of the starter dataset (4M images) for depth and surface normal estimation have state-of-the-art performance on the in-the-wild OASIS benchmark. To demonstrate the effectiveness of the pipeline for semantic tasks, we show that the predictions from a network trained for panoptic segmentation on a smaller 3-dataset portion (1M images) are of similar quality to models trained on COCO [32]. Full experimental details and more results are available in the supplementary.

Method	Test Data	L1 Error (↓)	$\delta > 1.25$ (↓)	$\delta > 1.25^2$ (↓)	$\delta > 1.25^3$ (↓)
XTC [63]	OASIS [12]	1.180	85.28	71.86	60.22
MiDaSv3 [41]		0.8057	82.03	67.25	55.35
`Omnidata`		**0.7901**	**81.00**	**65.22**	**52.93**
XTC [63]	NYU [47]	0.5279	70.41	49.90	36.28
MiDaSv3 [41]		0.3838	63.84	41.65	28.97
`Omnidata`		**0.2878**	**51.73**	**30.98**	**20.86**

Table 2: **Zero-shot depth estimation.** On NYU and OASIS, a DPT-Hybrid trained on the `Omnidata` starter dataset is comparable or better then the same model trained on existing depth datasets.

Method	Training Data	Angular Error° Mean	Angular Error° Median	% Within $t°$ 11.25°	% Within $t°$ 22.5°	% Within $t°$ 30°	Relative Normal AUC_o	Relative Normal AUC_p
Hourglass [11]	OASIS [12]	**23.91**	18.16	**31.23**	59.45	**71.77**	0.5913	0.5786
Hourglass [11]	SNOW [13]	31.35	26.97	13.98	40.20	56.03	0.5329	0.5016
Hourglass [11]	NYU [48]	35.32	29.21	14.23	37.72	51.31	0.5467	0.5132
PBRS [66]	NYU [48]	38.29	33.16	11.59	32.14	45.00	0.5669	0.5253
UNet [45]	SunCG [49]	35.42	28.70	12.31	38.51	52.15	0.5871	0.5318
UNet [45]	**`Omnidata`**	24.87	**18.04**	31.02	**59.53**	71.37	**0.6692**	**0.6758**
Human (Approx.)	-	17.27	12.92	44.36	76.16	85.24	0.8826	0.6514

Table 3: **Zero-shot surface normal estimation on OASIS.** A UNet trained on the `Omnidata` starter dataset matched or outperformed models trained on OASIS itself, and it matched human-level AUC_p. Notice that the first row is not zero-shot since it's trained on OASIS.

Monocular depth estimation: The current best approach for depth estimation is to aggregate multiple smaller datasets and train with scale- and shift-invariant losses [42, 41] to handle the different unknown depth ranges and scales. As of this writing, the DPT-based [41] models from "MiDaS v3.0" [41] define the state-of-the-art, especially on NYU [47]. We adopt a similar setting to MiDaS v3.0, but train on a 5-dataset portion of our starter dataset instead of their 10-dataset mix[1].

As in [41], we evaluate zero-shot cross-dataset transfer with test predictions and GT aligned in scale and shift in inverse-depth space. Tab. 2 shows that the DPT-Hybrid trained on our starter dataset outperforms MiDaS DPT-Hybrid on both the test set of NYU [47] and the validation split of OASIS (the test GT is not available). The error metrics use $\delta = max(\frac{d}{d^*}, \frac{d^*}{d})$ where d and d^* are aligned depth and ground truth. Our model better recovers the fine-grained details and true shape of the objects—this is especially clear in the surface normals extracted from the predictions (last 2 rows of Fig. 6). Full details, code, and more qualitative results are available on our website.

Surface normal estimation: Similar to the existing models on the surface normal track of OASIS, we train a vanilla UNet [45] architecture (6 down/6 up, similar to [63]) with angular & L1 losses, light 2D data augmentation, and input resolutions between 256 and 512. We use Adam [26] with LR 10^{-4} & weight decay 2×10^{-6}. The results in Tab. 3 indicate that our model matched human-level performance on OASIS AUC_p. On most of the remaining metrics, it outperformed related models trained on other datasets (including OASIS itself) and models with architectures specifically designed for normals estimation (PBRS). Fig. 7 shows that our model qualitatively performs *much* better on selected images than is indicated by the numbers, which may be because the standard metrics do not align with perceptual quality as "uninteresting" areas (walls, floors) dominate the

[1] MiDaS v3.0 also uses MTAN [33] for dataset balancing, and though in Sec. 5.3 we examine MTAN (it indeed helped on our dataset), we used here a naive sampling strategy in order to be consistent with the majority of the other models in this paper.

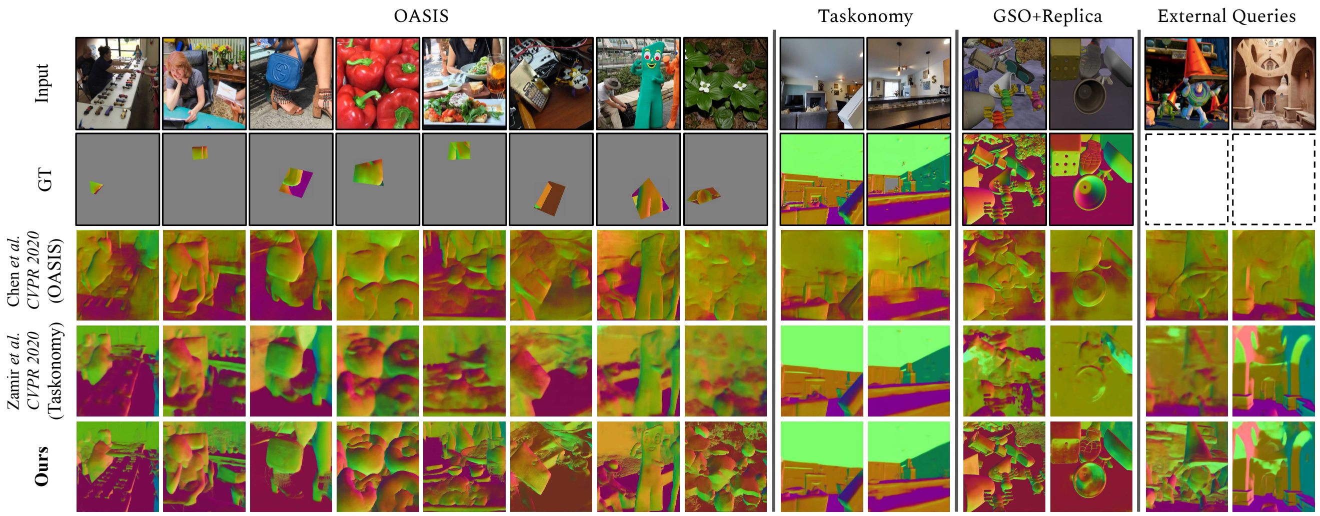

Figure 7: **Qualitative results of zero-shot surface normal estimation.** The 3 models are trained on OASIS[12], Full Taskonomy[63, 64], and our starter set. Queries are from 3 different datasets (OASIS, Taskonomy, GSO+Replica) in addition to some external queries in the last 2 columns (no ground truth available) which show the generalization of the models to external data [best viewed zoomed in].

score [12]. Further details and results are available in the supplementary.

Panoptic segmentation: To demonstrate the pipeline's ability to train models for non-geometric tasks, we train a PanopticFPN [27] on a 3-dataset subset of our starter dataset. Fig. 8 shows that on in-the-wild images of indoor buildings, the resulting model is of similar quality to one trained on COCO [32] (an extensive manually labeled dataset). Quantitative results, full experimental details, and code are available on our website.

Figure 8: **Qualitative results of panoptic segmentation with PanopticFPNs [27] trained on COCO [32] and `Omnidata`.** The `Omnidata` model trained jointly on Taskonomy, Replica, and Hypersim shows good out-of-distribution performance on indoor scenes without people.

4.3.1 Dataset Relatedness

To estimate how the components of the starter dataset are related, we use zero-shot cross-dataset transfer performance for surface normal and panoptic segmentation models trained on different components. Tab. 4 shows that each single model performs well on its corresponding test set, but typically generalizes poorly. The models trained on larger splits perform better overall (see supplementary). The model trained on the largest set achieved the best average performance (harmonic mean 25.8% and 30.3% better than best single-dataset models for surface normal estimation and panoptic segmentation). The ranking of transfers depended on the task, which might be due to the sparse panoptic labels on Taskonomy (from the followup paper [2]), but we believe the dependency is true in general.

	Surface normal estimation: L1 Error (↓)						Panoptic Quality (PQ) (↑)			
Train/Test	Taskonomy	Replica	Hypersim	Replica+GSO	BlendedMVG	h. mean	Taskonomy*	Replica	Hypersim	h. mean
Taskonomy*	**4.85**	7.76	8.69	13.89	15.55	8.53	8.39	3.95	11.67	6.55
Replica	9.36	**3.98**	11.78	10.28	15.02	8.24	1.01	**41.97**	4.50	2.43
Hypersim	7.28	7.57	**6.72**	11.34	12.94	8.56	**9.35**	14.08	25.39	13.80
Replica+GSO	13.88	4.94	15.05	**5.17**	14.03	8.26	-	-	-	-
BlendedMVG	17.1	14.23	16.93	14.87	**8.85**	13.58	-	-	-	-
`Omnidata`	5.32	4.24	6.53	6.45	11.53	**6.11**	9.14	41.24	**30.16**	**17.98**

Table 4: **Inter-dataset domain transfer performance for surface normal estimation and panoptic segmentation.** Models trained on each individual dataset and `Omnidata` are evaluated on test splits of the starter set. The harmonic mean across datasets is shown in the last column. (* PQ on *things* classes only, as Taskonomy does not feature *stuff* labels.)

5. Illustrative Data-Focused Analyses

Now that we have established that the annotator produces datasets capable of training reliable models, what analyses can we do with such datasets? We survey a few examples here, but they are not intended to be comprehensive (Sec. 1).

5.1. New 3D Data Augmentations

Data augmentation is used to address shortcomings in model performance and robustness. For example, models trained only on images captured with narrow apertures (e.g. NYU or Taskonomy) tend to perform poorly on images taken with a wide aperture (i.e. strong depth-of-field), and augmenting with 2D Gaussian blur is used to improve model performance on unfocused portions of images. The approach is common enough that 2D blur was included in the Common Corruptions benchmark [23]. Because the full scene geometry is available for our starter dataset, it is possible to do the *data augmentation in 3D* (image refocusing) instead of 2D (flat blurring). Fig. 9 shows an example of what 3D "image refocusing" augmentation on our dataset looks like. In the supplementary, we show that models trained for surface normal estimation using only 3D augmentation were more robust to both 2D blurring and 3D refocusing than those trained with 2D augmentation.

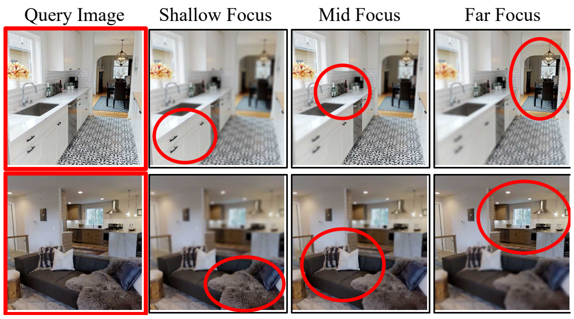

Figure 9: **Image refocusing augmentation on Taskonomy.** Portions of the image that are in focus are highlighted in red [best viewed zoomed in].

5.2. Mid-Level Cues as Inputs: Are They Useful?

Is there an advantage in using multiple sensors or non-RGB representations of the environment? Instead of predicting mid-level cues as the downstream task (i.e. multi-task learning), multiple cues could also be used as inputs (if relevant sensors are available) or specified as intermediate representation (with the labels being used as supervision only during training, i.e. PADNet [60]).

Tab. 5 demonstrates that using these additional cues in the latter 2 ways can improve performance on the original test set and also on unseen data. In this experiment, we train HRNet-18 [52] backbones for semantic segmentation using a single component dataset (10 spaces from Replica) and evaluate them on Replica, Hypersim and Taskonomy (tiny split). Relative to using only RGB inputs and the semantic segmentation labels, cross-entropy performance improves across the board when treating the cues as sensors (23%, 34% and 30%) or using them as intermediate representations (13%, 17%, and 19%). Adding more cues seems to help. Full experimental settings in the supplementary.

Future work could further analyze how the effectiveness of these different methods change with dataset size, which cues to use, how many additional images a mid-level cue is worth, and on the relative importance of getting more data from the same scene vs. adding data from new scenes.

Input/Supervised Domains	GT Mid-Level Cross-Entropy (↓)			Predicted Mid-Level Cross-Entropy (↓)		
	Repl.	H.Sim	Task.	Repl.	H.Sim	Task.
RGB	0.61	5.87	7.55	0.61	5.87	7.55
(All Above) + Normals	0.47	4.47	6.12	0.61	5.44	7.12
(All Above) + 3D Edges	**0.46**	4.47	6.75	0.54	5.06	6.49
(All Above) + (2D Edges, Z-Depth, 3D Keypts)	**0.46**	**3.86**	**6.04**	**0.53**	**4.9**	**6.13**

Table 5: **Utility of mid-level cues.** The table shows semantic segmentation results using models trained on Replica. The models (except for "RGB") received (either predicted or GT) mid-level cues in their input in addition to the RGB. The results show they notably benefited from the mid-level cues.

5.3. Systematic Evaluation of Multi-Task Learning

Recent work [55] shows that existing MTL techniques for computer vision appear to be specialized to their development setting, and in general they do not outperform single-task or shared-encoder approaches on novel datasets or tasks. We extend those results for additional tasks (3D Keypoints) and add numbers on our dataset as a comparison point. Specifically, we follow [55] and train models for a fixed set of tasks (semantic segmentation, 3D keypoints, depth z-buffer, and occlusion edges) using different MTL methods (Tab. 6). On a 3-dataset split of the starter dataset, some methods naturally perform better and others do worse. One might hope that the ordering of these methods would be the same on different tasks (semantic segmentation vs 3D Keypoints), or at least when training for those same tasks on a different dataset (NYU [47], CityScapes [17], or Taskonomy [64]). Yet, Tab. 6 shows that MTL methods display no clear ranking in either case (i.e. Spearman's ρ was always indistinguishable from 0). Ignoring the lack of significance, the cross-dataset correlation was still weak ($\rho < 0.45$), and methods performance was actually *anti-correlated* across tasks (ρ=-0.4), suggesting that the models are indeed specialized to specific tasks. This anti-correlation was true even when controlling for dataset.

Given that current MTL approaches do not outperform single-task baselines, predicting different mid-level cues poses a challenging setting for MTL. The `Omnidata` pipeline provides an avenue to create large and diverse multi-task mid-level benchmarks that could more systematically and reliably evaluate progress in multi-task learning.

Method	Semantic Segmentation: Ours IoU (↑)	Rank	NYU [47] IoU (↑)	Rank	CityScapes. [54] IoU (↑)	Rank	Taskonomy [54] IoU (↑)	Rank	3D Keypoints: Ours L1 (↓)	Rank	Taskonomy [54] L1 (↓)	Rank
Single task	85.12	1	90.69	2	65.2	1	43.5	4	0.0439	4	0.23	1
MTL baseline	81.82	3	90.63	3	61.5	4	47.8	1	0.0429	3	0.34	2
MTAN [33]	83.00	2	91.11	1	62.8	3	43.8	3	0.0426	1	0.4	3
Cross-stitch [37]	80.69	4	90.33	4	65.1	2	44.0	2	0.0427	2	0.50	4
Spearman's ρ	Within task.: ρ=0.43. Between segm.-3D keypts.: ρ=-0.4								Within task: ρ=0.2.			

Table 6: **Multi-task training methods do not show a clear ordering.** Within-task, rankings between different methods were indistinguishable from random orderings (i.e. ρ=0). Between tasks, rankings on Sem. Seg. were anti-correlated with rank on the 3D Keypts (ρ=-0.4). Both conclusions were strengthened after controlling for training setups.

6. Conclusion

This paper introduces a pipeline, `Omnidata` , to bridge the gap between real-world 3D scans and static vision datasets. The pipeline makes it possible to generate large and diverse datasets for many tasks with greater control over the data generation parameters, and to pursue fundamental lines of research such as how the data sampling strategy and collected cues impact model reliability. We examine a few example analyses, on a starter dataset generated from the pipeline. To demonstrate the utility and quality of the data, we trained a few standard vision models on this starter data and they achieved state-of-the-art performance on multiple computer vision tasks.

Acknowledgement: We thank Roman Bachmann for the discussions, implementations, and experimental help.

References

[1] Ignition app.

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[3] Iro Armeni, Sasha Sax, Amir Roshan Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.

[4] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[5] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[6] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[7] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Danny Gutfreund, Joshua Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. 2019.

[8] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

[9] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[11] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *CoRR*, abs/1604.03901, 2016.

[12] Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild, 2020.

[13] Weifeng Chen, Donglai Xiang, and Jia Deng. Surface normals in the wild. *CoRR*, abs/1704.02956, 2017.

[14] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, volume 2008, pages 129–136. Salerno, Italy, 2008.

[15] BO Community. Blender–a 3d modelling and rendering package. 2018.

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[19] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[20] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[21] J. Gibson. The senses considered as perceptual systems. 1966.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[24] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[25] Michal Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaskowski. Vizdoom: A doom-based AI research platform for visual reinforcement learning. *arXiv preprint arXiv:1605.02097*, 2016.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[27] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6392–6401, 2019.

[28] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[29] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[30] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

[31] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: a composite dataset for multi-domain

semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2020.

[32] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[33] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[35] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *CoRR*, abs/1806.08568, 2018.

[36] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[40] Senthil Purushwalkam Shiva Prakash and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *Advances in Neural Information Processing Systems*, 33, 2020.

[41] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[42] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[43] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[44] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. arXiv 2020.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[47] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[48] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. *Indoor Segmentation and Support Inference from RGBD Images*, pages 746–760. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[49] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[50] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[51] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, 2017.

[52] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[53] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

[54] Simon Vandenhende, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: Deciding what layers to share. *CoRR*, abs/1904.02920, 2019.

[55] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task net-

works: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019.

[56] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[57] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016.

[58] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[59] Fei Xia, Amir Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[60] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[61] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[62] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018.

[63] Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task consistency. *arXiv*, 2020.

[64] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[65] Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks, 2019.

[66] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas A. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *CoRR*, abs/1612.07429, 2016.