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

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

https://omnidata.vision

Abstract

The following items are provided in the supplementary material:

I. A live demo to run our networks on your own query images and a dataset design tool to visualize the effects of different dataset design parameters (§ 1).

II. Code, Docker, runnable examples and a documentation of usage for the annotator, tools, and the starter dataset (§ 2).

III. Mid-level cues provided by `Omnidata` annotator and their definitions (§ 3).

IV. Results of surface normal estimation with refocusing augmentation on blurred data (§ 4).

V. A description of GSO+Replica dataset generation process (§ 5).

VI. Dataset ablation analysis on surface normal estimation and panoptic segmentation for the starter set (§ 6).

VII. Visualization and evaluation of the "Blind Guess" (statistically informed guess) for the starter set (§ 7).

VIII. More qualitative results of surface normal estimation on OASIS dataset (§ 8).

IX. Full experimental setup for multi-task learning rank reversal experiment (§ 9).

1. Online Demos

The project website includes a live demo that allows to run our pretrained networks on your own uploaded query images. You can visualize the predictions for different tasks and see a comparison of `Omnidata` models to various baselines. The demo page also contains a link to the "demo archive" where you can browse uploads from other users. We also provide a dataset design tool that allows playing with different dataset design choices to visualize their effect on the sampled data.

2. Dockerized Pipeline, Tools, and Documentation

We provide a Dockerized Pipeline with all necessary software (Blender [10], MeshLab [9], and other libraries) installed, Pytorch dataloaders for loading the generated data and applying the necessary transforms for reading in each modality to analytic values, a starter dataset along with download scripts and other utilites. We also provide Omnidata Docs which includes a documentation on how to use all the open-sourced material of our paper.

3. Mid-level Cues Provided

This section describes the default mid-level cues and additional outputs provided by the `Omnidata` annotator.

3.1. 2D Cues

***2D Unsupervised Segmentation*:** Gestalt psychology proposes grouping as a primary mechanism through which humans learn to perceive the world as a set of coherent objects [33]. The annotator provides groupings based on normalized cuts [28] of the RGB image into perceptually similar spatially coherent groups.

Texture Edges: offer low-level cues about object boundaries. Classic computer vision pipelines commonly use edges as an intermediate representation in a larger processing pipeline. The annotator provides edges from a Canny [5] edge detector without nonmax suppression.

2D Keypoints: are designed to indicate possibly important pixels and identify them across images. These are frequent in both vision [14] and robotics [22, 23] pipelines. The annotator provides pre-nonmax-suppression SURF intensity maps to identify potentially important regions of the RGB image, and the fragments cue (see below) can be used to link points across images.

3.2. Single-View 3D Cues

***Depth: Z-Buffer*:** For each pixel, the metric distance from the point to the camera plane. The most common form of depth in computer vision + robotics.

***Depth: Euclidean*:** For each pixel, the metric distance from the point to the camera's optical center. This can be used (e.g.) for adding lens blur (Sec. 6.2 of the main paper).

***Surface Normals*:** Crucial for computer vision and robotics tasks (e.g. for computing lighting, grasp estimation, etc.): the tangent vector relative to the camera of the corresponding point on the mesh.

Principal Curvature: For each pixel, the principal curvatures κ_1 and κ_2, which are also sufficient for computing Gaussian ($\kappa_1 \cdot \kappa_2$) and mean ($(\kappa_1 + \kappa_2)/2$) curvature. These quantities are invariant under rigid transformations, and curvature is known to be important in primate visual processing [34].

***Occlusion Edges*:** indicate boundaries where one pixel occludes something behind it. While 2D edges respond to to changes in texture, 3D edge features depend only on 3D geometry and are invariant to color and lighting.

***(re)Shading*:** One cue to infer scene geometry from an RGB image is "shape from shading" [3] via the intrinsic image decomposition $I = A \cdot S$ into an albedo A and a shading function S parameterized by lighting and depth. The decomposition is thought to be useful for human visual perception [2]. We define a (re)shading cue for S as follows: Given an RGB image, the label is the shading function S that results from having a single point light at the camera origin, and S is multiplied by a constant fixed albedo A.

***3D Keypoints*:** 3D keypoints, like 2D, are designed to indicate possibly important points and link them across viewpoints. Unlike 2D keypoints, 3D keypoints are often designed to be invariant to informative (but possibly distracting) cues such as texture [39, 30, 24, 35, 17]. Based on its specificity and robustness, we use the pre-nonmax-suppressed output of [30] for this cue.

***2.5D Unsupervised Segmentation*:** uses the same graphcut algorithm as 2D, but the labels are computed jointly from the RGB, depth image, and surface normals. Thus the 2.5D segmentation cue incorporates information about scene geometry that is not present in the RGB image but readily inferred by humans.

***Manhattan Vanishing Points*:** Vanishing points offer useful information about the scene geometry [25, 18], particularly a "Manhattan world" [11, 38, 4] with three dominant vanishing points (X, Y, and Z axis). We provide the X, Y, and Z Manhattan vanishing points (Gaussian sphere format).

***Camera Intrinsics*:** Deep networks are excessively sensitive to changes in camera intrinsics such as field-of-view. We provide camera intrinsics for each image.

3.3. Multi-View 3D Cues

***Camera Extrinsics*:** provides camera RT matrices for each image.

***Point Matching*:** indicates which other preselected points are present in this view. Useful for point matching tasks such as [14].

Fragments (Optical Flow): Each space_point_view image contains an image whose pixel values encode the corresponding mesh face, and these values are consistent across images in the space and can be decoded to approximate global 3D coordinates or used for optical flow. This would be akin to perfect feature descriptors for either 2D or 3D keypoints.

3.4. Semantic Cues

If the dataset supports, the annotator can provide cues for the following:

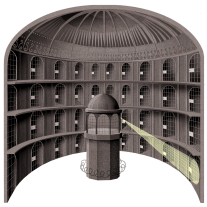
***Class Presence*:** labels provide a present/not present indicator used for image classification.

***Instances*:** identify the instance identity of each pixel. Regardless of class, this gives an object-centric grouping.

***Semantic Class*:** the semantic category for each pixel.

***Panoptic Segmentation*[1]:** combination of semantic segmentation and instance identification [16].

[1]The *Panopticon* (from 18th-century philosopher Jeremy Bentham) was conceived as the ideal prison; an institutional system of control-by-surveillance whereby a centralized security guardhouse can observe all prisoners in one view, while subjects are unable to tell whether they are being watched. Though instantiated as a building, Bentham intended it as a method for any institution, with the threat of observation, to force compliance and docility. Expanded and popularized by Foucault [13] in the 20th century, *Panopticism* remains influential among disciplines across the humanities and social sciences. The *panopticon* is also well-known in popular culture; as "Big Brother" in the surveillance narrative *Nineteen Eighty-Four*, for example, and by name in the cover story of the most recent edition of *The Economist* ("The People's Panopticon," *7 Aug. 2021 edition*).



An illustrated panopticon. Two of the first 10 images returned for the Google query *"modern panopticon"* make reference to Facebook, including one that is simply the above image with the Facebook logo superimposed.

Since Facebook is dealing with public reprobation over its transgressive surveillance policies and (repeated) privacy violations, it probably did not intend for the designation *panoptic segmentation* to bear such an unfortunate resemblance to *panopticism*. In any case, a concrete name like "Per-Pixel Category and Instance Classification" would be clearer and less provokative.

3.5. Additional Information

***RGB*:** RGB images can be real image scans if provided, or they can be rendered from the textured mesh.

Masks that indicate whether the pixel corresponds to an area missing from the mesh.

4. Surface Normal Estimation with Refocusing Augmentation

As described in Sec. 6.2 of the main paper, the mid-level cues can be used as data augmentations in addition to training targets. While defocus cue can be useful in depth estimation [15, 6, 29], we explore it as refocusing augmentation on our dataset, which is possible due to availability of camera parameters and euclidean depth. We provide quantitative and qualitative surface normal estimation results for training with this augmentation. Tab. 1 compares 2 models trained with and without this augmentation evaluated on both refocused and blurred test data from our starter set. We use Gaussian blur with kernel size 3 and sigma uniformly chosen in the range (0.1, 2). As shown by the results, the model trained with refocusing augmentation shows much better performance on the blurred data. The gap is clearer as shown by images in Fig. 1. The figure shows that the baseline model would easily fail with a small amount of blur present in the input, and the refocusing augmentation has a substantial effect in increasing the robustness to these blur effects. We repeat the experiment with different levels of blur in the input using different kernel sizes (3, 5, 7, 9). Fig. 2 shows the performance of the models for each amount of blur. As the plots show, the model trained with augmentation shows good performance even for high levels of blur, while the accuracy drops significantly in the baseline model as the blur increases.

Refocusing Augmentation	Test Data	Error (↓) L1	Error (↓) MSE	Angular Error° (↓) Mean	Angular Error° (↓) Median	% Within t° (↑) 11.25°	% Within t° (↑) 22.5°	% Within t° (↑) 30°
✗	Blurred	7.54	2.04	16.86	8	61.91	75.73	81.17
✓		**6.44**	**1.61**	**14.37**	**6.40**	**66.20**	**79.31**	**84.53**
✗	Refocused	6.45	1.63	14.42	6.52	66.36	79.55	84.67
✓		**6.14**	**1.48**	**13.685**	**6.108**	**67.23**	**80.46**	**85.64**

Table 1: **Surface normal estimation with refocusing augmentation.** The models are evaluated on blurred and refocused test split of the starter set. Gaussian blur with kernel size 3 is used for blurring the input. As the results show, refocusing augmentation improves the performance of the model on blurred data.

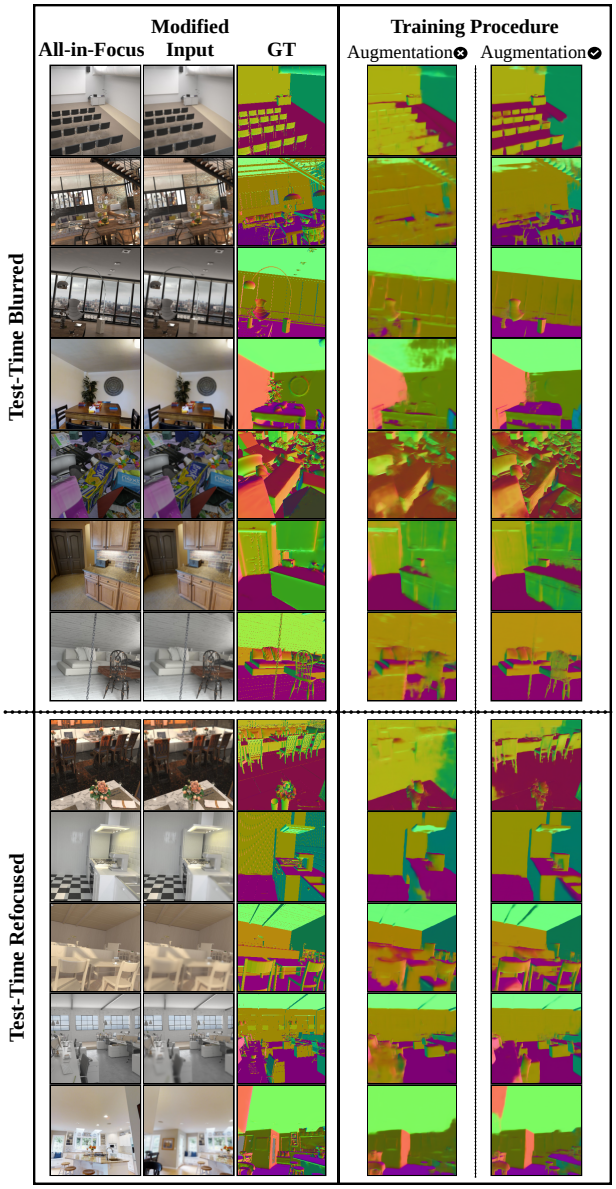


Figure 1: **Qualitative results for refocusing augmentation.** The results compare the models trained with and without refocusing augmentation on both refocused and blurred data from the test splits of the starter set. Same parameters as training are used for refocusing the test data. We also use Gaussian blur with kernel size 3 for blurring the input. Clearly the model trained with the augmentation shows much more robustness to blur effects while the baseline model easily fails with a small amount of blur [best viewed zoomed in].

5. GSO+Replica Dataset Generation Process

We scatter Google Scanned Objects [1] around Replica [31] buildings to create object-centric views. Habitat [21] environment is used to generate physically plausible scenes. The dataset is provided in 3 different object densities for each space (3, 6, 15 objects per square meter which we refer to as low, medium, high density). Objects are randomly sampled from 1032 objects provided in Google Scanned Objects, and they are scattered uniformly across the building according to the density. To create object-centric views, thousands of cameras are generated in each space using Poisson Disc Sampling. Points of interest are only sampled from the objects rather than the whole mesh. For each point-of-interest, a subset of cameras with an unobstructed line-of-sight of the point are selected. Cameras

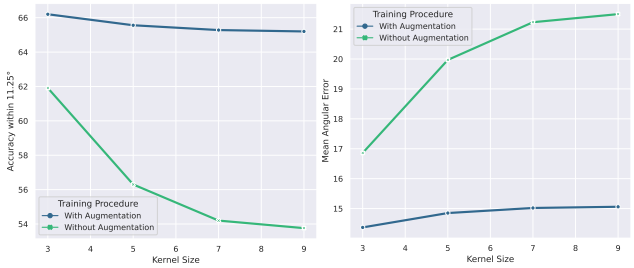


Figure 2: **Performance of the 2 training procedures (w/ and w/o refocusing augmentation) for different amounts of blur in test data.** We use Gaussian blur with kernel sizes 3, 5, 7, and 9, to produce different amounts of blur in the input. The plots show the "Accuracy within 11.25°" (left) and "Mean Angular Error" (right) for the 2 models for each kernel size. It is shown that the performance of the baseline model significantly drops with increasing the amount of blur while the model trained with augmentation shows much more robustness.

are filtered according to an additional constraint so that the point-camera distance is between 0.2 and 1 meter to make sure we have an object-centric view. The views are saved for each camera-point combination in which the camera is fixated on the point-of-interest. Examples of images from each object density are shown in Fig. 3.

6. Dataset Ablation Analysis of the Starter Set

To assess the contribution of each single dataset in our starter set, we list the zero-shot transfer performance to OASIS [8] and COCO [19] for models trained on each single dataset of the starter set, and some combinations of them.

The results listed in Tab. 2 and 3 provide an understanding of the impact of each dataset component in our starter set. Models trained on only scene-level data such as Taskonomy [37], Hypersim [27], or Replica [31] result on poor performance on objects, while the model trained on GSO+Replica will have an object-centric bias with poor performance on backgrounds and scenes. We provide a starter dataset with both scene- and object-centric views which, as shown by the results, is necessary for final best performance and generalization to in-the-wild data. Furthermore, including all datasets is necessary since the diversity present in the whole starter dataset will further improve the generalization.

7. Blind Guesses (Statistically Informed Guesses) for the Starter Set

Similar to [36], we compute the blind guesses (query-agnostic statistically informed guess) from the starter set for each domain. We evaluate the blind guess for surface normals on OASIS data, and the test split of the starter set in Tab. 4. The reported results will provide an estimation of the lower bound performance for these datasets. We also compare our blind guesses to the ones computed only from the Taskonomy dataset. A visualization of these guesses for surface normals and reshading are provided in Fig. 4. Comparing the blind guesses for the 2 datasets demonstrates that there is less bias present in the starter set compared to the Taskonomy alone, such as the ceiling bias present in the top part of the image for surface normal blind guess of Taskonomy which is not the case in our starter set. Better performance of the starter set blind guess (compared to the Taskonomy alone) on OASIS data, as shown in Tab. 4, will further prove the point.

8. Surface Normal Estimation on OASIS Dataset

In this section, we include additional qualitative results from our surface normal estimation experiments on OASIS [8]. Fig. 5 qualitatively compares the models trained on Full Taskonomy and the starter set on some sample images from the val split of OASIS. As shown by the figure, the model trained on Full Taskonomy has poor performance on objects and largely misses the details as opposed to the model trained on the starter set.

9. Multi-Task Learning Rank Reversal Experimental Setup

In this section, we explain the experimental setup for the multi-task learning rank reversal experiment provided in the section 5.3 of the main paper. Similar to [32], we use a simple shared-encoder MTL model, a single task baseline, as well as 2 other common MTL approaches (MTAN [20] and Cross-stitch [26]) for our experiment. Each encoder is a ResNet-50 model with dilated convolutions and pre-trained on ImageNet [12]. We use Deeplab [7] head for the task specific decoders. Each multi-task model is trained on the 4 following tasks: semantic segmentation, 3D keypoints, depth z-buffer, and occlusion edges. We use medium Taskonomy, Replica, and Hypersim as the training data, and evaluate the models performance on semantic segmentation and 3D keypoints on tiny Taskonomy test set. Table 4 of the main paper provides the results for this experiment.

References

[1] Ignition app. 3

[2] E. H. Adelson and A. P. Pentland. The perception of shading and reflectance. *Perception as Baysian Inference*, pages pp. 409–423, 1996. 2

[3] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, Aug 2015. 2

[4] J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645, June 2012. 2

Figure 3: **Sample images from GSO + Replica dataset.** Images shown are from the 3 different object densities (3, 6, 15 objects/m^2) included in GSO+Replica dataset.

Training Data					Anglular Error° (↓)		% Within t° (↑)			Relative Normal (↑)	
Taskonomy	Replica	Hypersim	Replica+GSO	BlendedMVG	Mean	Median	11.25°	22.5°	30°	AUC_o	AUC_p
✓					29.68	21.78	25.73	51.29	62.66	0.6220	0.6163
	✓				33.06	26.03	19.03	43.60	56.31	0.5711	0.6099
		✓			29.94	22.81	21.64	49.36	62.28	0.6375	0.6311
			✓		33.22	26.46	15.81	41.97	56.21	0.5669	0.5893
				✓	29.77	23.23	23.47	48.64	61.23	0.5661	0.6033
✓	✓				28.62	21.59	24.01	51.85	64.40	0.6260	0.6248
✓	✓	✓			28.61	21.55	23.81	51.94	64.78	0.6614	0.6566
✓	✓	✓	✓		27.73	20.43	25.72	54.06	66.51	**0.6686**	0.6596
✓	✓	✓	✓	✓	**26.34**	**19.39**	**28.66**	**56.37**	**68.43**	0.6572	**0.6832**

Table 2: **Zero-shot transfer performance on OASIS dataset.** Models are evaluated on val split of OASIS dataset. The results show the impact of each single dataset in the starter set on the performance of surface normal estimation and generalization to in-the-wild data.

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 1

[6] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. Deep depth from defocus: how can defocus blur improve 3d estimation using dense neural networks? In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 3

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 4

[8] 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. 4

[9] 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. 1

[10] BO Community. Blender–a 3d modelling and rendering package. 2018. 1

[11] James M. Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 845–851. MIT Press, 2001. 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[13] Michel. Foucault and Alan Sheridan. *Discipline and punish : the birth of the prison / Michel Foucault ; translated from the French by Alan Sheridan*. Penguin Harmondsworth, 1979. 2

[14] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Kosecka. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. *CoRR*, abs/1802.07869, 2018. 1, 2

[15] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7683–7692, 2019. 3

[16] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018. 2

[17] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. *Hough Transform and 3D SURF for Robust Three Dimensional Classification*, pages 589–602. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 2

Training Data			Taskonomy			Replica						Hypersim					
Taskonomy	Replica	Hypersim	$PQ_{th}(\uparrow)$	$SQ_{th}(\uparrow)$	$RQ_{th}(\uparrow)$	$PQ_{th}(\uparrow)$	$SQ_{th}(\uparrow)$	$RQ_{th}(\uparrow)$	$PQ_{st}(\uparrow)$	$SQ_{st}(\uparrow)$	$RQ_{st}(\uparrow)$	$PQ_{th}(\uparrow)$	$SQ_{th}(\uparrow)$	$RQ_{th}(\uparrow)$	$PQ_{st}(\uparrow)$	$SQ_{st}(\uparrow)$	$RQ_{st}(\uparrow)$
✓			8.39	38.88	9.54	6.99	37.67	9.32	-	-	-	21.76	68.46	26.94	-	-	-
	✓		1.01	17.39	1.31	28.82	56.45	36.01	**55.12**	69.72	**65.08**	2.89	35.68	3.90	6.11	28.95	8.99
		✓	9.35	54.25	**11.90**	14.67	55.56	18.65	13.48	29.73	18.57	23.90	65.73	29.94	26.87	52.29	35.14
✓	✓		**10.27**	45.85	11.82	28.66	57.87	35.98	54.17	**70.18**	63.76	8.90	38.29	11.04	10.25	27.02	14.61
✓		✓	8.70	40.67	10.13	9.44	50.56	12.07	15.37	33.35	20.61	26.28	67.64	32.43	30.72	54.03	38.97
	✓	✓	9.09	**61.48**	11.69	**44.07**	**75.94**	**53.88**	48.99	64.00	58.10	24.67	68.51	30.46	19.04	37.52	24.68
✓	✓	✓	9.14	41.95	10.29	30.14	57.92	37.50	52.35	64.87	61.67	**27.79**	**68.86**	**34.28**	**32.53**	**54.77**	**40.84**

Table 3: **Ablation of training datasets for panoptic segmentation.** Transfers to and from Taskonomy only evaluate *things* labels, as Taskonomy does not feature any stuff labels.

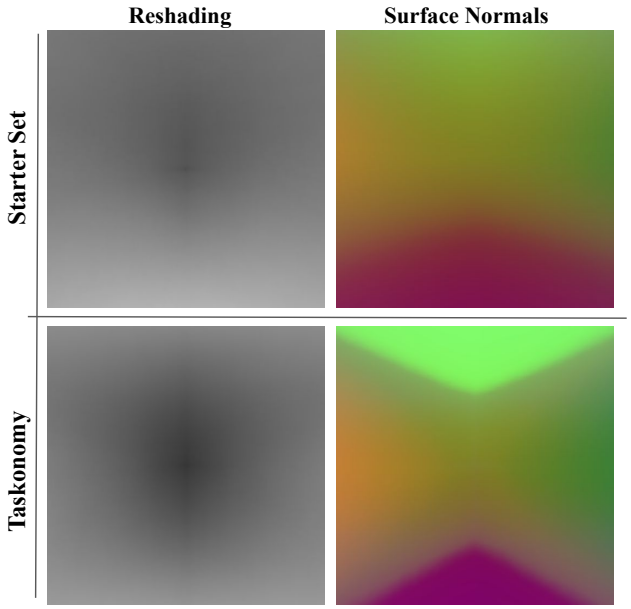


Figure 4: **Statistically informed guesses ("Blind Guess") on the Starter Set.** Blind guesses computed from the starter set and Taskonomy alone are shown for 2 domains. Comparing the surface normal blind guess for the 2 datasets will show that there is less bias present in our starter set comparing to Taskonomy alone (the ceiling bias which is only present in Taskonomy blind guess).

		Angular Error° (↓)		% Within t° (↑)			Relative Normal (↑)	
Test Data	Blind Guess	Mean	Median	11.25°	22.5°	30°	AUC_o	AUC_p
OASIS	Starter Set	**35.28**	**30.64**	**14.48**	**36.7**	**49.03**	**0.5352**	0.4302
	Taskonomy	41.73	35.80	10.28	29.00	41.38	0.5282	**0.4404**
Starter Set	Starter Set	**43.72**	**43.04**	7.41	21.6	**32.17**	-	-
	Taskonomy	44.88	44.66	**8.87**	**22.57**	31.97	-	-

Table 4: **Blind guess evaluation on OASIS and starter set.** The blind guesses computed from our starter set and Taskonomy alone are evaluated on val split of OASIS and test split of the starter set. The results will provide a lower bound for performance on these benchmarks.

[18] Hui Kong, J. Y. Audibert, and J. Ponce. Vanishing point detection for road detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 96–103, June 2009. 2

[19] 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. 4

[20] 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. 4

[21] 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. 3

[22] Lucas Manuelli, Wei Gao, Peter R. Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *CoRR*, abs/1903.06684, 2019. 1

[23] Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning, 2020. 1

[24] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2):348–361, Sep 2010. 2

[25] O. Miksik. Rapid vanishing point estimation for general road detection. In *2012 IEEE International Conference on Robotics and Automation*, pages 4844–4849, May 2012. 2

[26] 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. 4

[27] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. arXiv 2020. 4

[28] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000. 1

[29] Pratul P Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T Barron. Aperture supervision for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6393–6401, 2018. 3

[30] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 44, 2010. 2

[31] 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

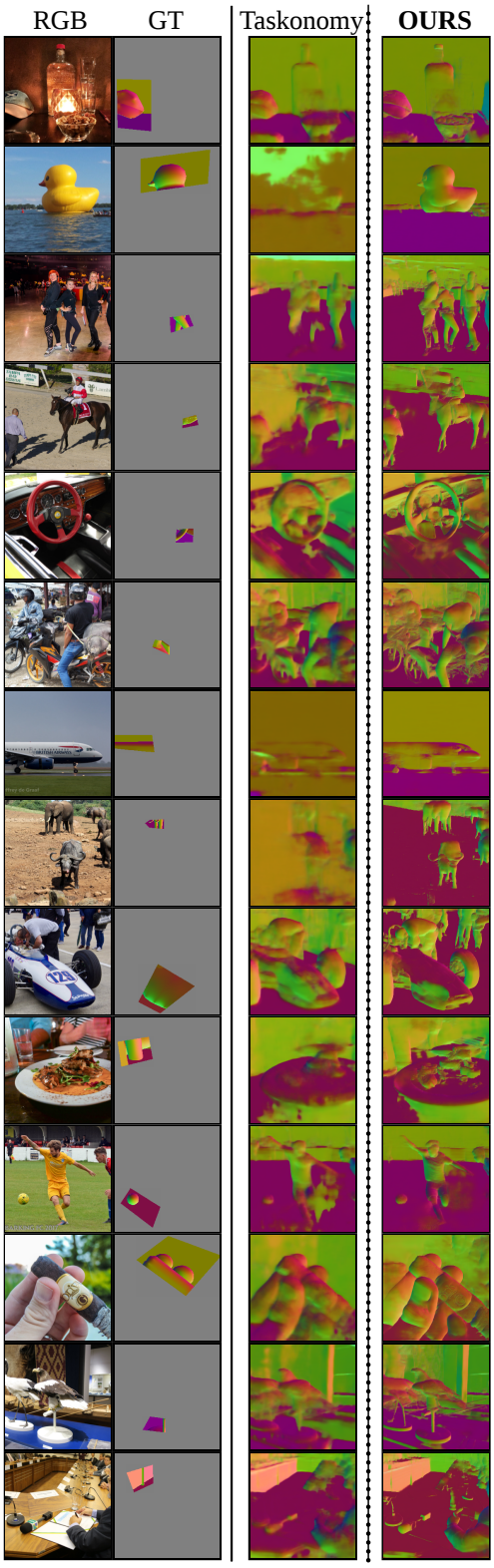


Figure 5: **Qualitative results on OASIS data.** The 2 models are trained on Full Taskonomy and the starter set. The Taskonomy model has poor performance on objects and largely misses the details.

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. 3, 4

[32] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019. 4

[33] Max Wertheimer. Laws of organization in perceptual forms. *Psycologische Forschung*, 4:301–350, 1923. 1

[34] Xiaomin Yue, Irene S. Pourladian, Roger B. H. Tootell, and Leslie G. Ungerleider. Curvature-processing network in macaque visual cortex. *Proceedings of the National Academy of Sciences*, 111(33):E3467–E3475, 2014. 2

[35] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–380, June 2009. 2

[36] 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. 4

[37] 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. 4

[38] Lilian Zhang, Huimin Lu, Xiaoping Hu, and Reinhard Koch. Vanishing point estimation and line classification in a manhattan world with a unifying camera model. *International Journal of Computer Vision*, 117(2):111–130, Apr 2016. 2

[39] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696, Sept 2009. 2