Supplementary Material for *Occlusion Boundary and Depth: Mutual Enhancement via Multi-Task Learning*

Lintao XU[1] Yinghao WANG[2] Chaohui WANG[1†]

[1] LIGM, Univ Gustave Eiffel, École des Ponts, CNRS, France

[2] INFRES, Télécom Paris, Institute Polytechnique de Paris, France

Our Supplementary Material is organized as follows:

- **Section A.1** presents the mathematical formulation of our multi-task loss function;
- **Section A.2** details the datasets used in the main paper and in the supplementary material, including synthetic datasets, real-world datasets, and cross-domain real-world test sets.
- **Section A.3** documents implementation specifics, training schedules, hardware configurations *etc*.
- **Section B.1** explains the procedural generation pipeline of our diverse, photorealistic *OB-Hypersim* dataset.
- **Section B.2** presents a comparison of *OB-Hypersim* with existing self-occlusion–handled OB benchmarks.
- **Section C.1** provides additional quantitative comparisons with state-of-the-art depth-only methods and their ability to handle OBs.
- **Section C.2** provides additional ablations on the proposed *OBDCL*.
- **Section C.3** provides additional quantitative zero-shot comparisons and *MoDOT* results on outdoor scenes.
- **Section C.4** and **Section C.5** supplements additional quantitative analyses and addition studies.
- **Section D** provides further qualitative visualizations.

This structure ensures systematic technical reproducibility while offering detailed insights beyond the constraints of the main paper.

A. Further Methodological and Experimental Details

A.1. The Multi-Task Loss Functions

The overall loss function used to train the proposed multi-task (MT) method *MoDOT* is formulated as follows:

$$\mathcal{L} = w_d \cdot \mathcal{L}_{\mathcal{D}} + w_{ob} \cdot \mathcal{L}_{\mathcal{OB}} + w_c \cdot \mathcal{L}_{\mathcal{C}} \tag{1}$$

For depth estimation loss, given a Ground-Truth (GT) depth map with K pixels having valid depth values, the Scale-Invariant Logarithmic (SILog) loss [3] $\mathcal{L}_{\mathcal{D}}$ is computed as follows:

$$\mathcal{L}_{\mathcal{D}} = \alpha_d \sqrt{\frac{1}{K}\sum_i {\Delta d_i}^2 - \frac{\lambda_d}{K}\left(\sum_i \Delta d_i\right)^2}, \quad \Delta d_i = \log \hat{d}_i - \log d_i^* \tag{2}$$

where λ_d is a variance minimizing factor, α_d is a scale constant, and d_i^* represent the GT depth value and $\hat{d}_i$ is the predicted depth at pixel i. Following previous work [41], λ_d is set to 0.85 and α_d is set to 10 in the experiments.

For Occlusion Boundary (OB) estimation loss, given an input image I, the binary GT OB map $\mathcal{B}$, and denotes four side outputs and a final probability map obtained during training as $\hat{\mathcal{B}_{1-5}}$, where $\mathcal{B}, \hat{\mathcal{B}}_i = (b_j, j = 1, ..., |\mathcal{I}|), b_j \in (0, 1)$, we define $\mathcal{B}_-$ and $\mathcal{B}_+$ as the sets of boundary and non-boundary pixels, respectively. The total loss $\mathcal{L}_{\mathcal{OB}}$ for OBs and Class-Balanced Cross-Entropy (CCE) loss [4] Γ can be formulated as:

$$\mathcal{L}_{\mathcal{OB}} = \sum_{i=1}^{5}(w_i \cdot \Gamma(\hat{\mathcal{B}}_i, \mathcal{B})), \quad \Gamma(\hat{\mathcal{B}}, \mathcal{B}) = -\lambda_b \alpha_b \sum_{j \in (\mathcal{B}_-)} \log(1 - \hat{b_j}) - (1 - \alpha_b) \sum_{j \in (\mathcal{B}_+)} \log(\hat{b_j}). \tag{3}$$

Where w_i denote the loss weights for four intermediate side outputs and the final OB prediction. Here, $\alpha_b = \frac{|\mathcal{B}_+|}{|\mathcal{B}_+|+|\mathcal{B}_-|}$ balances the boundary/non-boundary pixels, and λ_b controls the weights of the positive samples over negative ones. In our experiments, the values w_i were primarily set to $0.1, 0.3, 0.3, 0.5, 2.3$, and λ_b was set to 1.1 for all datasets training, except for OB-FUTURE [31], where λ_b was adjusted to 1.7.

The formulation of the proposed OB-Depth Constraint Loss (*OBDCL*) L_C is presented in the main paper, so we omit its details here for brevity. *OBDCL* is theoretically sound and has proven effective in the ablation study of the main paper and also in this Supplementary Material. However, in practice, the GT depth annotations are often imprecise—especially for subtle depth differences such as those between a poster and the wall behind it. These cases correspond to true OBs but are not accurately reflected in the depth GT. Since this constraint introduces a strong supervisory signal, we apply it with a small loss weight (*i.e.*, 0.1) to prevent it from overwhelming the gradient flow during backpropagation.

A.2. Complete Dataset Introduction

In this section, we provide a more comprehensive introduction to the datasets used in the experiments in the main paper and the Supplementary Material. Additional details about the proposed *OB-Hypersim* are stated in Section B.

- **OB-FUTURE** is built using the 3D definition-based OB generation method proposed in [31], applied to 3D indoor scenes from the *3D-FUTURE* dataset [5]. This OB generation process is directly derived from the mathematical definition in [26], enabling more accurate OB annotations that include complete self-occlusions. Compared to OB-FUTURE, *OB-Hypersim* offers a more diverse set of photorealistic scenes, although the OB annotations are pseudo and less precise. OB-FUTURE consists of 17,267 training images and 1,869 test images, each with a resolution of 1080×1080.
- **NYUD-v2** dataset [22] (for MT learning) includes a diverse range of indoor scenes, such as offices and living rooms *etc*., with 795 training images and 654 testing images at a resolution of 640×480 (the valid mask resolution is 560×425 pixels for both training and testing phases.). It provides several dense annotations, including monocular depth and object boundaries. As object boundaries constitute a subset of OBs, these annotations effectively offer partial OB labeling. Therefore, in our work, the object boundaries in NYUD-v2 are interpreted as partial OB annotations for training and testing. For single-task monocular depth estimation settings (*e.g.*, [41], the models in their paper are typically trained on the raw NYUD-v2 dataset using approximately 30,000 samples. In contrast, multi-task learning settings often rely on a smaller, curated subset of the dataset containing annotations for both tasks.
- ***NYUv2-OC++*** [18] includes 654 indoor images with a valid resolution of (592×440), sourced from all the NYUD-v2 dataset's testing images [21]. Its GT OBs were labeled based on visible depth discontinuities. In our experimental setup in the Supplementary Material, we explore an unconventional setting where we reverse the standard NYUD-v2 configuration: We use the imperfect OB annotations from the NYUv2-OC++ test set as ground-truth OB labels with GT depth for multi-task learning, while evaluating the depth predictions on the original NYUD-v2 training set.
- **iBims-1**[8] (Independent Benchmark Images and Matched Scans – Version 1) is a high-quality RGB-D dataset specifically designed to evaluate single-image depth estimation methods. We report qualitative and quantitative results using models trained solely on *OB-Hypersim* to assess their zero-shot cross-domain transferability without any domain adaptation and fine-tuning.
- **DIODE** [25] is a standard high-resolution RGB-D dataset commonly used for monocular depth estimation. We provide zero-shot quantitative results on this dataset, covering both indoor and outdoor scenes, to demonstrate our method's ability to capture structures in unseen environments.
  OB-DIODE is a 50-image subset of the DIODE dataset with manually annotated, self-occlusion–handled OBs, introduced in [31]. We evaluate zero-shot OB performance on this subset.
- **OB-EntitySeg** is a 70-image subset of the EntitySeg dataset [15] with manually annotated, self-occlusion–handled OBs, introduced in [31]. We also evaluate zero-shot OB performance on this subset.

A.3. Implementation Details

Our method is implemented in PyTorch and trained on NVIDIA RTX A5000 GPUs. The networks (both stages one and two) are optimized end-to-end using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use a default batch size of 8 and employ a learning rate schedule that decays from its initial value to 10% over time. We adopt a two-stage training strategy: In the first stage, following prior work [4, 28, 31], we crop input images to 320×320 and train the models from scratch with a learning rate of 3×10^{-5} for all datasets. In the Second Stage Refinement (*SSR*), we train the stage two models using full-resolution images specific to each dataset while keeping the stage-one parameters frozen. We apply additional data augmentations inspired by multi-task learning methods [24, 38] throughout both training stages. The training configurations are dataset-specific:

- OB-FUTURE: 20 epochs (stage one) / 10 epochs (*SSR*) with *SSR* learning rate 5×10^{-5}.
- *OB-Hypersim*: 50,000 iterations (stage one) / 20,000 iterations (*SSR*) with *SSR* learning rate 3×10^{-4}.
- NYUD-v2: 50 epochs per stage using full-size images (batch size 4 in stage one) with *SSR* learning rate 3×10^{-3}.

The stage one network produces depth maps at $\frac{1}{4}$ spatial resolution (relative to input) which are upsampled to the original resolution via a PyTorch interpolation layer. In contrast, the stage one OB maps maintain full input resolution. During Stage Two *SSR*), both depth and OB predictions are generated directly at full input resolution. In addition, during stage one training on *OB-Hypersim*, *OBDCL* is applied after 20,000 iterations.

Following depth-only methods (*e.g.*, [1, 35, 41]), we use a sigmoid layer at the end of the depth decoder to map predictions to the range [0,1]. These values are then rescaled to metric depth by multiplying by the dataset's maximum depth for *OB-Hypersim*, and to the range [0,255] for OB-FUTURE.

B. *OB-Hypersim* Details

B.1. *OB-Hypersim* OB Generation and Processing

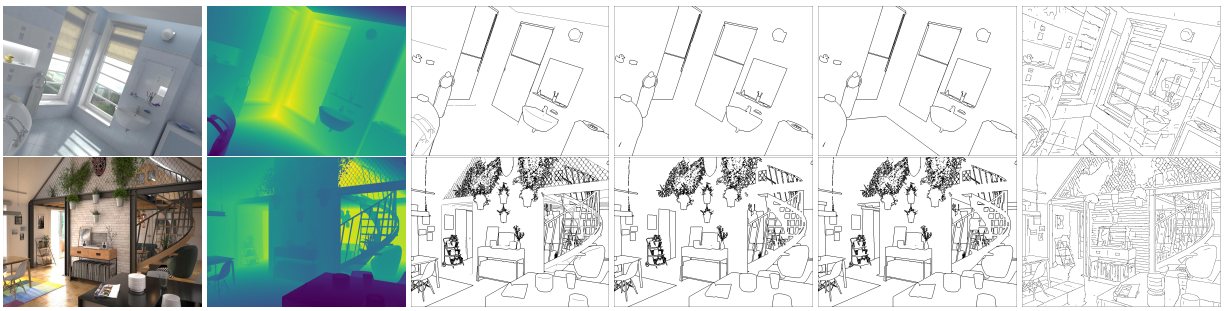

Figure 1. **Comparison of GT annotations in *OB-Hypersim* used for the ablation study (Table 3 in the main paper).** From left to right: RGB image, depth GT map, our generated pseudo OBs, instance contours,segmentation contours, and pseudo edges.

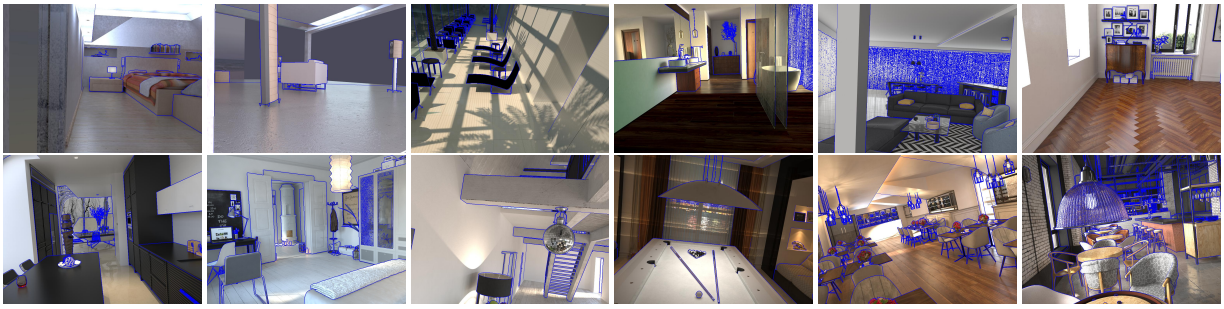

Figure 2. **Visualization of images removed due to noisy ground-truth OBs.** The pseudo-OB annotations (blue) are superimposed on the original RGB image. Significant noise is observable in multiple scene regions—including walls, tables, ceilings, floors, and lighting fixtures—where inaccurate OB annotations are produced.

The GT OBs in our proposed *OB-Hypersim* were mainly generated using the 2D occlusion-simulation based method in P2ORM [16]. *OB-Hypersim* sourced from the Hypersim dataset [19]—a photorealistic synthetic dataset designed for holistic indoor scene understanding. Hypersim provides 77,400 images across 461 diverse indoor scenes, each with detailed dense per-pixel labels and ground truth geometry. Due to the wide variation and complexity of scene geometry, the depth across different regions varies significantly.

P2ORM detects OBs by first identifying 2D depth discontinuities—regardless of occlusion presence—under a fixed depth difference threshold. However, using a single global threshold for all scenes is suboptimal given the dataset's diversity. Therefore, we apply dynamic thresholding to adaptively generate OB maps with P2ORM. Since P2ORM often produces incomplete contours and fails to capture all OBs [31], we supplement the P2ORM-generated OBs by integrating instance segmentation contours to create more comprehensive pseudo ground truth OBs. Models trained on our proposed *OB-Hypersim* dataset exhibit better zero-shot generalization performance on real-world data compared to those trained on synthetic OB-FUTURE.

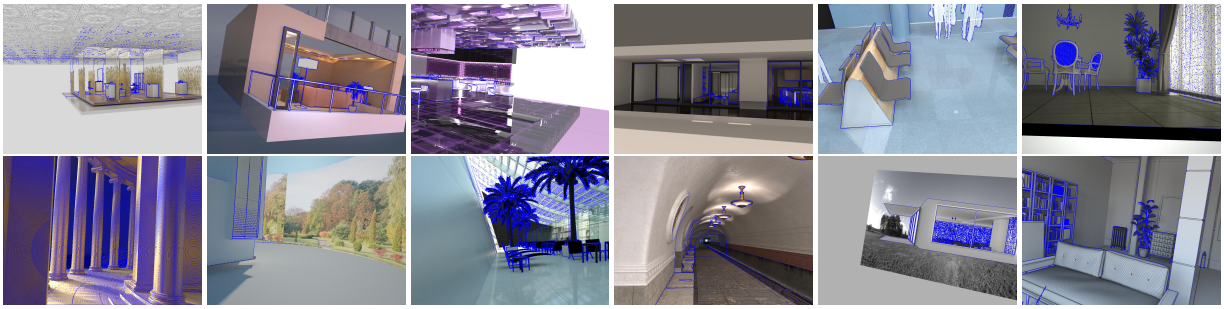

Figure 3. **Visualization of removed inappropriate scenes examples.** These scenes were excluded due to issues such as unrealistic layouts, incomplete geometry, or severe noise in the ground truth annotations.

Additionally, due to the noise introduced by P2ORM—especially in distant regions of the images—and the presence of unsuitable scenes in Hypersim, we manually filtered out a significant number of noisy or low-quality samples. Examples of such noise and discarded scenes are shown in Fig. 2 and Fig. 3.

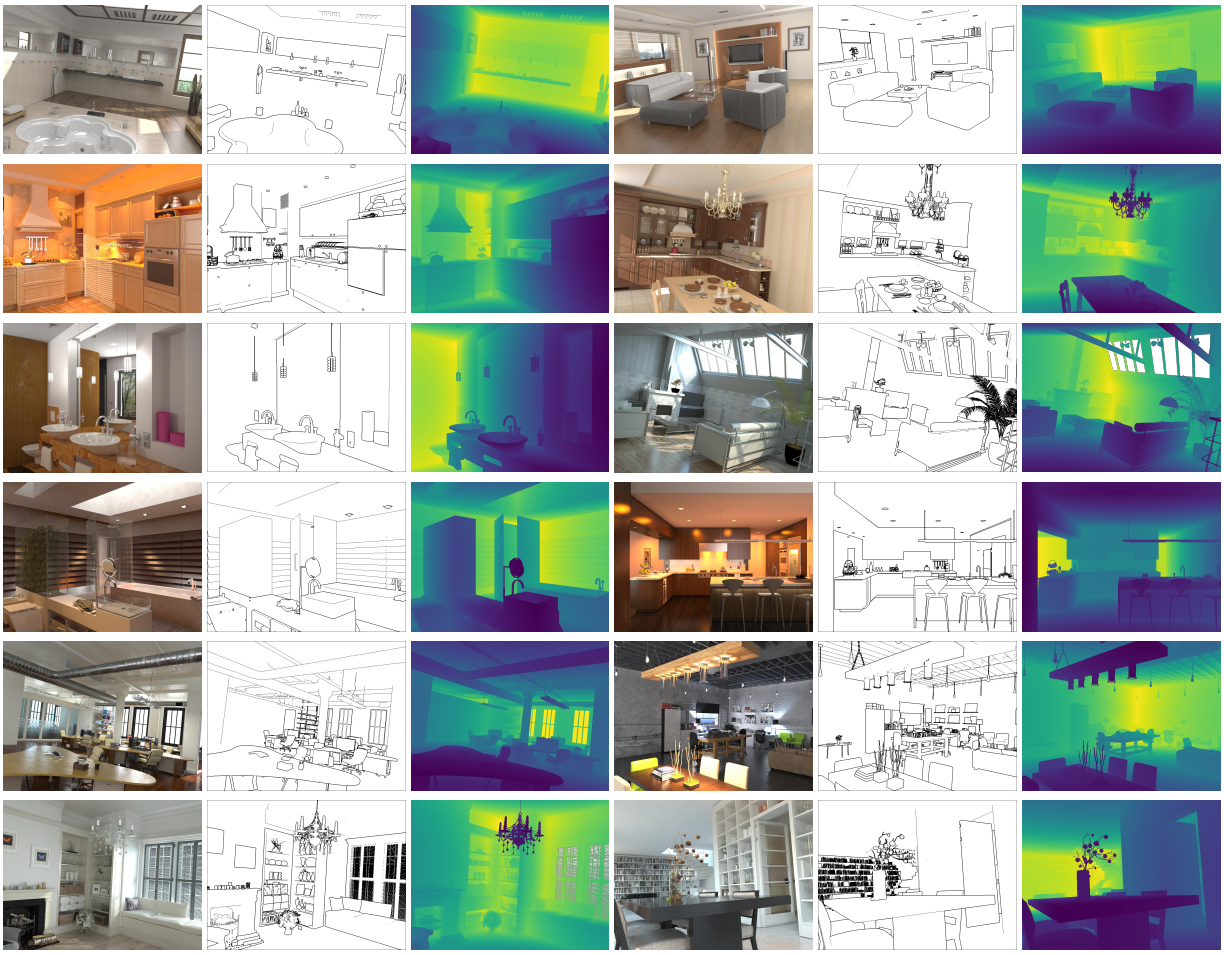

Figure 4. **More visualizations of the scenes from final *OB-Hypersim***. Example scenes are shown along with their corresponding ground truth OBs and depth maps.

After the filtering process, the final resulting dataset includes 254 scenes for training, comprising 27,536 images, and 30 scenes for testing, with 3,311 images. Each scene includes 1 to 8 camera viewpoints capturing the scene from different

angles, with all images rendered at a resolution of 1024×768. Further data visualizations on more remained scenes are provided in Fig. 4.

B.2. Dataset Comparison

Occlusion patterns are usually more complex in outdoor scenes (*e.g.*, grass, plants), prior works [16, 31] have mainly focused on indoor scenarios. Similarly, our *OB-Hypersim* also targets indoor scenes. A dataset comparison is shown in Table 1. Compared with OB-FUTURE, *OB-Hypersim* provides more diverse and complex indoor environments (*e.g.*, conference rooms, bedrooms, offices, bathrooms, restaurants, kitchens, study rooms *etc*.), whereas OB-FUTURE includes only bedrooms with varying furniture types.

For OB GT, OB-FUTURE is less noisy since it is generated directly from 3D meshes and definition-based methods. Although our *OB-Hypersim* removes most the noisy OB ground truth, some residual noise and incomplete boundaries remain.

For GT depth, OB-FUTURE provides Blender-generated relative depth in the range [0, 255], where the added backgrounds are assigned a depth of 255 and treated as invalid masks during depth supervision. This added background makes the depth discontinuous and the scenes less realistic. In contrast, we use the metric depth from Hypersim's HDF5 source files for training and testing. These depth maps are continuous, and the scenes appear more realistic.

Table 1. **Comparison with existing self-occlusion–handled OB benchmarks and NYUD-v2.** All datasets are indoor scenes. Our *OB-Hypersim* is the largest and most diverse in the community, and the OB ground-truth quality remains high despite some minor noise.

Dataset	Volume	Resolution	Scene	GT Annotation	GT Quality	Self-occlusion	With Depth
OB-FUTURE [31]	19,186	(1080, 1080)	synthetic	synthetic	high	✓	✓
NYUv2-OC++ [16, 18]	654	(592, 440)	real	manual	low (incomplete)	✓	✓
iBims1_OR [16]	100	(640, 480)	real	synthetic	low (incomplete)	✓	✓
InteriorNet_OR [16]	10,000	(640, 480)	synthetic	synthetic	low (incomplete)	✓	✓
NYUD-v2 [22]	1,449	(640, 480)	real	manual	middle (incomplete)	×	✓
OB-DIODE [31]	50	(1024, 768)	real	manual	high	✓	✓
OB-EntitySeg [31]	70	(982, 882)	real	manual	high	✓	×
Our *OB-Hypersim*	30,847	(1024, 768)	photorealistic	synthetic	high	✓	✓

C. More Experimental Details

C.1. Comparison with State-of-the-Art Depth-Only Methods

The objective of this work is to explore two closely connected aspects of 3D scene perception—depth and OBs—their mutual relationship and whether they can benefit each other through multi-task learning. Although the goal is not to develop a state-of-the-art depth estimator, it is still interesting to examine the performance of representative depth-only methods, particularly their ability to extract occlusion boundaries.

We compare *MoDOT* with two depth foundation models, Depth Anything [34] and Depth Anything V2 [35], evaluating their depth and OB extraction performance on synthetic datasets OB-FUTURE and *OB-Hypersim*. Following prior works [16, 17], we use the Canny operator (skimage) to extract depth edges as coarse OBs.

Table 2. **Ablation on different Canny thresholds for extracting coarse OBs from depth maps for OB evaluation.**

		T (0.05, 0.15)		T (0.15, 0.30)		T (0.25, 0.50)	
Method	**Dataset**	**OB-Recall↑**	**OB-Fscore↑**	**OB-Recall↑**	**OB-Fscore↑**	**OB-Recall↑**	**OB-Fscore↑**
Depth Anything Large	OB-FUTURE	0.0846	0.1276	0.0581	0.0928	0.0386	0.0646
Depth Anything v2 Large	OB-FUTURE	**0.1136**	**0.1630**	0.0873	0.1323	0.0642	0.1015
Depth Anything Large	*OB-Hypersim*	0.0294	0.0446	0.0198	0.0341	0.0133	0.0238
Depth Anything v2 Large	*OB-Hypersim*	**0.0595**	**0.0936**	0.0491	0.0809	0.0402	0.0681

As shown in Table 2, we test three threshold pairs for the Canny edge detector: (0.05, 0.15), (0.15, 0.30), and (0.25, 0.50). Note that the established baseline from prior works [16, 17] uses (0.15, 0.30). In skimage, these thresholds represent fractions of the maximum gradient magnitude after normalizing the depth map to [0,1]. For example, with threshold (0.15,0.30),

- Gradients $>30\%$ of the maximum are strong edges.
- Gradients $\geq 15\%$ (but $\leq 30\%$) are retained only if connected to strong edges.

Table 3. **Comparison with Depth Anything and Depth Anything v2 on OB-FUTURE (relative depth).** We use their released relative-depth pretrained models, directly evaluated on the OB-FUTURE test set with GT depth normalized to [0,1] and Canny thresholds set to (0.05, 0.15).

Method	OB-Recall↑	OB-Fscore↑	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑
Depth Anything Large	0.0846	0.1276	0.3524	0.9329	0.9021	0.4215	0.3487	0.1549
Depth Anything Base	0.0953	0.1427	0.3503	0.8922	0.9042	0.4236	0.3346	0.1596
Depth Anything Small	0.0916	0.1372	0.3423	0.8077	0.8921	0.4149	0.3047	0.1749
Depth Anything v2 Large	0.1136	0.1630	0.3547	0.8981	0.9177	0.4369	0.3349	0.1627
Depth Anything v2 Base	0.1140	0.1637	0.3562	0.8825	0.9147	0.4345	0.3298	0.1660
Depth Anything v2 Small	0.1291	0.1852	0.3478	0.8383	0.9073	0.4280	0.3155	0.1684
Ours	0.9090	**0.6131**	0.0396	0.1020	0.0901	0.0052	0.0380	0.9427
Ours + *SSR*	**0.9486**	0.5415	**0.0381**	**0.0974**	**0.0843**	**0.0047**	**0.0361**	**0.9518**

Table 4. **Comparison with depth-only methods on *OB-Hypersim* (metrics depth).** We use their released indoor metric-depth pretrained models, evaluated on the *OB-Hypersim* test set with metrics GT depth in ([0, 100]) and Canny thresholds set to (0.05, 0.15). For Depth Anything, only one large version metric-depth model (finetuned on NYUD-v2) is publicly available and their official zero-transfer report on the *OB-Hypersim* source data Hypersim [19] test set gives Abs Rel = 0.363 and δ_1 = 0.361. For Depth Anything v2, we use their released pre-trained indoor metric-depth models, which directly finetuned on Hypersim (60K).

Method	OB-Recall↑	OB-Fscore↑	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑
Depth Anything Large	0.0294	0.0446	1.1461	0.6358	0.4118	0.5581	0.2439	0.2677
Depth Anything v2 Large	0.0595	0.0936	1.1269	0.5934	0.3604	0.5260	0.2205	0.3433
Depth Anything v2 Base	0.0585	0.0911	1.1287	0.5984	0.3683	0.5287	0.2229	0.3307
Depth Anything v2 Small	0.0530	0.0824	1.1308	0.6034	0.3775	0.5346	0.2252	0.3249
Ours	0.8670	**0.5163**	0.6583	0.3463	0.2963	0.2279	**0.1223**	**0.5167**
Ours + *SSR*	**0.8732**	0.5109	**0.6537**	**0.3456**	**0.2954**	**0.2266**	0.1243	0.5148

In Table 3 and Table 4, we evaluate the performance of Depth Anything and Depth Anything v2 on two synthetic datasets. In Table 4, for their released indoor metric-depth pretrained models: Depth Anything is fine-tuned on NYUD-v2 (their official zero-transfer report on the *OB-Hypersim* source dataset Hypersim [19] gives Abs Rel = 0.363 and δ_1 = 0.361), while Depth Anything v2 is fine-tuned on Hypersim [19] (60K).

In Section C.3 Table 9, additionally, we present zero-shot results of our models (trained solely on *OB-Hypersim*) and Depth Anything on iBims-1.

Several interesting observations can be made from these quantitative results:

1. The depth comparison between *MoDOT* and Depth Anything on iBims-1 and *OB-Hypersim* shows that our depth performance is comparable, and in some metrics even better—especially considering the training scale. Depth Anything uses 1.5M labeled and 62M unlabeled images for training, Depth Anything v2 uses 595k labeled and 62M unlabeled images for training, while our *MoDOT* is trained on only 27K samples for 50,000 / 70,000 (stage two, SSR) iterations;
2. The two depth foundation models perform relatively poorly on OB-FUTURE, likely due to the unrealistic backgrounds leading to less reliable depth estimation (as discussed in Section B.2);
3. Stronger depth-only methods tend to yield better extracted OB evaluation scores, which might suggest a mutual relationship between OB and depth that supports our idea.

C.2. Further Exploration on *OBDCL*

OBDCL directly encodes the geometric consistency between OB and depth discontinuities, making it simple but highly demanding on ground truth. To further examine its performance and benefits, in Table 5 we conduct an ablation study using the full stage-one model (note that the ablation in the main paper was not on the full model) across the mainly used three datasets: OB-FUTURE (17k for training) contains simple synthetic scenes with high-quality GT; *OB-Hypersim* (27k) provides more diverse and complex photorealistic scenes with minor noise but still high-quality GT; and NYUD-v2 (795) consists of real scenes with incomplete OB (see dataset comparison in Sec. B.2. The quantitative results in Table 5 show that although these datasets differ in scale and GT quality, using *OBDCL* still benefits most depth and OB evaluation metrics.

Further, we explore the trade-off between *OBDCL* and dataset size. To quantify this, we conduct a dataset-scaling ablation: our stage-one models are trained with and without *OBDCL* at multiple training sizes (*e.g.*, 10%, 30%, 50%, 70%) on OB-

Table 5. **Ablation of *OBDCL* on datasets of different scales and GT quality.** Better results in each block are underlined.

Dataset	**OBDCL**	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
NYUD-v2 (795)	×	0.4195	0.1484	0.1200	0.0708	0.0502	0.8715	0.6295	0.1718
NYUD-v2 (795)	✓	0.4174	0.1475	0.1169	0.0692	0.0498	0.8741	0.6287	0.1729
OB-FUTURE (17K)	×	0.4044	0.1024	0.0898	0.0519	0.0380	0.9453	0.9338	0.5723
OB-FUTURE (17K)	✓	0.3963	0.1020	0.0901	0.0523	0.0380	0.9427	0.9090	0.6131
OB-Hypersim (27K)	×	0.6707	0.3584	0.3256	0.2483	0.1280	0.4974	0.8549	0.5244
OB-Hypersim (27K)	✓	0.6583	0.3463	0.2963	0.2279	0.1235	0.5167	0.8670	0.5163

Table 6. **Ablation of *OBDCL* on OB-FUTURE with varying dataset scales.** Better results in each block are underlined.

Data Scale	***OBDCL***	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
10%	×	0.4928	0.1251	0.1106	0.0821	0.0470	0.8937	0.9222	0.6058
10%	✓	0.4897	0.1249	0.1135	0.0838	0.0473	0.8916	0.9114	0.6159
20%	×	0.4546	0.1158	0.1028	0.0693	0.0435	0.9162	0.9213	0.5988
20%	✓	0.4491	0.1141	0.1005	0.0668	0.0426	0.9195	0.9235	0.5935
30%	×	0.4519	0.1162	0.0987	0.0636	0.0431	0.9190	0.9261	0.5848
30%	✓	0.4446	0.1148	0.0971	0.0616	0.0426	0.9213	0.9231	0.5906
40%	×	0.4367	0.1126	0.1005	0.0637	0.0420	0.9247	0.9102	0.6130
40%	✓	0.4230	0.1090	0.0947	0.0595	0.0402	0.9326	0.9247	0.6016
50%	×	0.4578	0.1199	0.1089	0.0711	0.0450	0.9059	0.9267	0.5824
50%	✓	0.4369	0.1138	0.1020	0.0638	0.0425	0.9192	0.9248	0.5924
60%	×	0.4220	0.1092	0.0968	0.0590	0.0406	0.9319	0.9462	0.5611
60%	✓	0.4293	0.1108	0.0985	0.0616	0.0413	0.9236	0.9287	0.5780
70%	×	0.4311	0.1102	0.0968	0.0592	0.0413	0.9305	0.9231	0.5930
70%	✓	0.4209	0.1076	0.0936	0.0562	0.0401	0.9335	0.9282	0.5932
80%	×	0.4282	0.1089	0.0931	0.0564	0.0404	0.9327	0.9323	0.5838
80%	✓	0.4179	0.1085	0.0957	0.0571	0.0402	0.9310	0.9307	0.5793
90%	×	0.4095	0.1062	0.0947	0.0560	0.0396	0.9363	0.9351	0.5687
90%	✓	0.4078	0.1051	0.0931	0.0541	0.0390	0.9404	0.9153	0.6055
100%	×	0.4044	0.1024	0.0898	0.0519	0.0380	0.9453	0.9338	0.5723
100%	✓	0.3963	0.1020	0.0901	0.0523	0.0380	0.9427	0.9090	0.6131

Table 7. Ablation of *OBDCL* on *OB-Hypersim* with varying dataset scales. Better results in each block are underlined.

Data Scale	***OBDCL***	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
10%	×	0.8772	0.4825	0.3903	0.3866	0.1758	0.3841	0.8030	0.4724
10%	✓	0.8570	0.4689	0.4003	0.3829	0.1706	0.3929	0.8202	0.4547
30%	×	0.7816	0.4204	0.3856	0.3509	0.1509	0.4251	0.8175	0.5020
30%	✓	0.7785	0.4269	0.3756	0.3357	0.1535	0.4343	0.8342	0.4878
50%	×	0.7677	0.4092	0.3767	0.3379	0.1485	0.4214	0.8633	0.4908
50%	✓	0.7520	0.3962	0.3566	0.3688	0.1430	0.4522	0.8401	0.5185
70%	×	0.7218	0.3834	0.3374	0.2765	0.1380	0.4713	0.8734	0.5057
70%	✓	0.6985	0.3710	0.3336	0.2660	0.1328	0.4908	0.8483	0.5260
100%	×	0.6707	0.3584	0.3256	0.2483	0.1280	0.4974	0.8549	0.5244
100%	✓	0.6583	0.3463	0.2963	0.2279	0.1235	0.5167	0.8670	0.5163

FUTURE and *OB-Hypersim*, which both have high-quality self-occlusion handled OB, and we evaluate depth and OB metrics versus training size. Training iterations are kept the same as in full-scale training, and test sets are fixed to better isolate performance changes. Results are reported in Table 6 and Table 7.

Our quantitative ablation studies above demonstrate that *OBDCL* remains effective across most different dataset sizes and maintains robustness to noisy/incomplete OB ground truth in most scenarios. While we acknowledge two current limitations—(1) *OBDCL* 's effectiveness may diminish for extremely large-scale training, and (2) optimal performance requires high-quality GT—we emphasize that no larger dataset currently exists beyond our processed *OB-Hypersim*(30k samples) that provides paired depth maps with self-occlusion handled OB annotations. This dataset already required significant curation to construct from available sources. Crucially, when using all currently accessible OB-labeled data, our approach shows consistent benefits for joint depth estimation and OB recognition. We recognize that in an idealized future with massive perfect datasets, alternative methods might surpass *OBDCL*, but in today's practical research context—where comprehensive OB annotations remain scarce—our method delivers measurable improvements.

C.3. More zero-shot results of *MoDOT*

In this section, we present additional zero-shot results of *MoDOT*. We first report quantitative performance on self-occlusion–handled OBs in OB-DIODE and OB-EntitySeg. Although these two datasets provide high-resolution high-quality manual annotations, their limited size (see Table 1) prevents full training and evaluation as in NYUD-v2. Therefore, we only present zero-shot cross-domain comparisons on these real datasets in Table 8. For the fixed OB-Fscore and OB-Recall used in this paper, in transfer evaluations, the fixed threshold is set to 0.5 instead of 0.7 in the in-domain evaluation, which increases recall but lowers F-score. We also report the classic MATLAB-based ODS/OIS/AP metrics used in prior edge/OB works [4, 23, 27, 31]: (i) *Fixed contour threshold (ODS)*, which is the F-measure with the best fixed OB probability threshold over the all datasets; (ii) *Best threshold of image (OIS)*, which is F-measure with the best OB probability threshold for each image; (iii) *Average precision (AP)*, which is the average precision over all occlusion probability thresholds. In addition, as discussed, two transformer-based models [36, 39] cannot be fully trained and tested on our *OB-Hypersim* due to inherent shape issues; thus, their quantitative results are not included in this section.

Table 8. **Zero-shot quantitative results on real-world self-occlusion–included OB datasets.**

	OB-EntitySeg					OB-DIODE				
Method	**OB-Recall↑**	**OB-Fscore↑**	**ODS↑**	**OIS↑**	**AP↑**	**OB-Recall↑**	**OB-Fscore↑**	**ODS↑**	**OIS↑**	**AP↑**
OB Baseline (ours)	0.6377	**0.3281**	69.5	71.0	51.9	0.0212	0.0397	71.0	72.5	56.4
SharpNet [17]	0.7308	0.2615	59.8	62.3	48.4	0.0337	0.0616	53.7	64.0	49.4
MTAN [12]	0.6823	0.1822	55.1	58.8	35.7	0.0393	0.0721	56.5	59.8	44.2
PAD-Net [30]	0.6663	0.1819	56.8	61.0	38.5	**0.0457**	**0.0838**	55.3	58.8	41.4
MTI-Net [24]	0.7449	0.2284	63.9	68.0	50.0	0.0355	0.0625	**71.9**	73.4	**65.8**
InvPT [38]	0.1238	0.0447	30.7	30.8	9.2	0.0226	0.0402	26.8	25.1	10.9
DenseMTL [13]	0.7708	0.3133	**69.7**	**72.9**	**64.6**	0.0252	0.0487	68.4	71.8	60.8
Ours	0.7707	0.3115	68.3	70.9	55.3	0.0265	0.0498	70.7	**73.9**	62.1
Ours + *SSR*	**0.7832**	0.3060	67.6	70.4	56.6	0.0280	0.0525	69.9	73.6	62.1

Several interesting observations can be drawn from Table 8: (1). The MT competitors struggle in zero-shot depth evaluation (see Table 9), their OB zero-shot results are comparable and in some cases even achieve the best scores. While our zero-shot OB results are not always the best, they remain comparable to the top-performing methods in most cases. (2). All models perform poorly on OB-DIODE under fixed Recall and F-score evaluation, which may indicate a significant domain shift; (3). Higher fixed Recall and F-score do not necessarily imply higher ODS/OIS/AP. (4). InvPT fails on the zero-shot setting, possibly due to positional embedding issues *etc*. Although we attempted to resolve the shape mismatch by interpolating the positional embeddings at test time (similar to [11]), the problem persisted.

In Table 9, we present the quantitative zero-shot results of *MoDOT* on the real-world dataset iBims-1. Compared to multi-task competitors, our method achieves superior performance and noticeably improved depth predictions; Compared to single-task baselines, joint training improves all depth and OB metrics, further validating the benefits of jointly learning depth and OB; Compared to Depth Anything, which is trained on 63.5M samples, our transfer depth performance remains comparable while obtaining better OB-Recall. Some multi-task competitors perform poorly in zero-shot transfer and even produce invalid results (*e.g*., SharpNet); we omit these extreme metrics from the table. Additionally, P2ORM [16] provides self-occlusion–handled OB GT for iBims-1 (named ibims_OR in Table 1). Since their GT quality is low (lacking complete object contours *etc*., one example can be found in Fig. 9), we report OB-Recall and omit the OB-Fscore evaluation on their OB GT for reference.

One benefit of jointly learning OB and depth is that the zero-shot results demonstrate a stronger ability to capture scene

Table 9. **Zero-shot results on iBims-1.** We report Depth Anything published depth metrics in their paper for iBims-1. And depth baseline and Depth Anything method's OB-Recall are computed with coarse OB extracted from depth predictions using the canny operator, as described in Section C.1.

Method	OB-Recall↑	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑
Depth Baseline [41]	0.0366	1.4371	0.4214	0.4458	0.7888	0.1579	0.3874
OB Baseline (ours)	0.7225	-	-	-	-	-	-
Depth Anything Large [34]	0.1249	-	-	**0.150**	-	-	**0.714**
MTAN [12]	**0.8450**	1.9705	0.7761	0.4768	0.9923	0.3056	0.0773
PAD-Net [30]	0.8280	2.0273	0.7593	0.4783	1.0525	0.3092	0.0866
MTI-Net [24]	0.8280	1.9809	0.7363	0.4811	1.0076	0.3059	0.0614
InvPT [38]	0.2574	2.0656	0.7191	0.4446	1.0478	0.2863	0.1367
DenseMTL [13]	0.7226	2.0276	0.7822	0.4903	1.0734	0.3231	0.0624
Ours	0.7807	**0.7702**	0.2308	0.1977	**0.1914**	0.0840	0.6747
Ours + *SSR*	0.7830	0.7739	**0.2305**	0.1929	0.1950	**0.0834**	0.6822

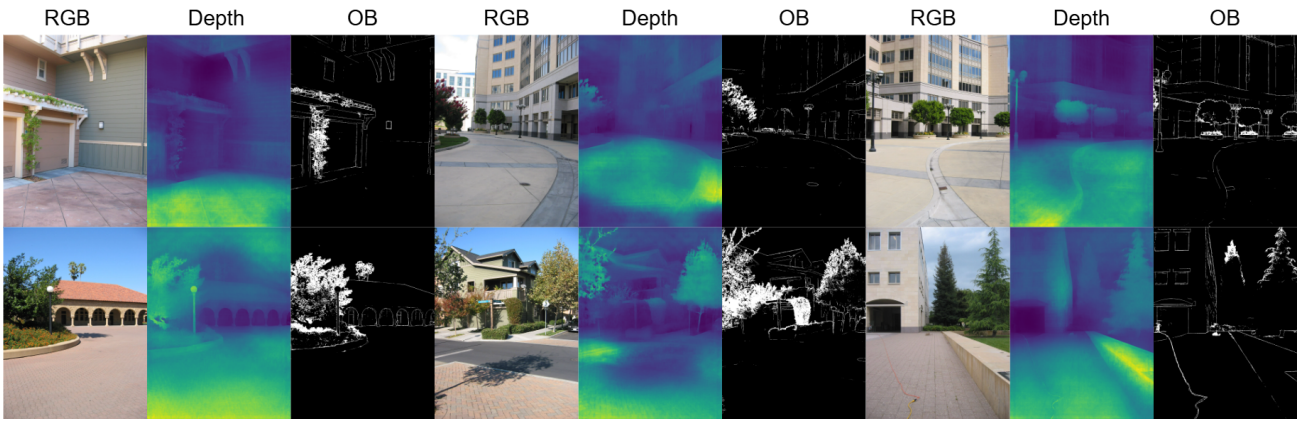

Figure 5. Additional qualitative zero-shot results of *MoDOT* on the outdoor scene on Make3D [20] dataset.

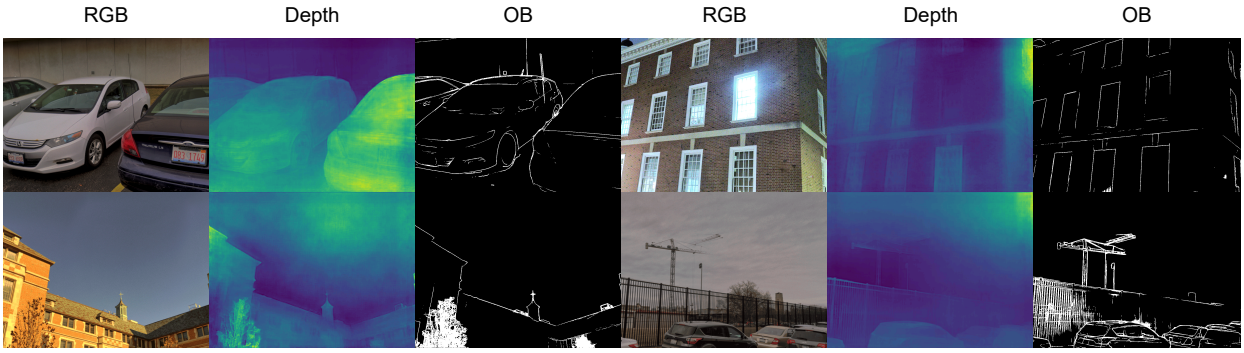

Figure 6. Additional qualitative zero-shot results *MoDOT* on the outdoor scene on DIODE [25] dataset.

geometry structure compared to single-task baselines and multi-task competitors. We provide qualitative zero-shot results of *MoDOT* on two outdoor depth datasets, Make3D [20] in Fig. 5 and DIODE-outdoor [25] in Fig. 6. Although *MoDOT* does not perform as well on these indoor datasets (*e.g.*, iBims-1), which may be due to the large domain gap between indoor and outdoor scenes, synthetic and real data, as well as biases in the GT distributions. Moreover, compared to indoor scenes, outdoor environments exhibit more complex occlusion patterns (*e.g.*, plants, trees, grass, and small-scale self-occlusions).

C.4. Experiments Supplementing the Main Paper

In Table 10 and Table 11, we complement the quantitative experiments in the main paper by presenting a full comparison of depth and OB metrics on two synthetic datasets used.

Table 10. **Full quantitative comparisons on the synthetic OB-FUTURE.**

Method	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
Depth Baseline [41]	0.4524	0.1149	0.1011	0.0655	0.0327	0.9215	-	-
OB Baseline (ours)	-	-	-	-	-	-	0.7655	0.5634
SharpNet [17]	0.9535	0.2492	0.1890	0.2475	0.0942	0.5966	**0.9571**	0.4005
MTAN [12]	0.5576	0.1619	0.1218	0.0993	0.0552	0.8523	0.9238	0.3537
PAD-Net [30]	0.5447	0.1406	0.1188	0.0932	0.0513	0.8627	0.9022	0.3348
MTI-Net [24]	0.5064	0.1295	0.1106	0.0800	0.0477	0.8891	0.9125	0.4000
InvPT [38]	0.9335	0.2425	0.2371	0.2997	0.0909	0.6122	0.3228	0.1678
DenseMTL [13]	0.5217	0.1321	0.1106	0.0842	0.0488	0.8818	0.8927	0.6030
Ours	0.3963	0.1020	0.0901	0.0523	0.0380	0.9427	0.9090	**0.6131**
Ours + *SSR*	**0.3809**	**0.0974**	**0.0843**	**0.0468**	**0.0361**	**0.9518**	0.9486	0.5415

Notably, as discussed in the main paper, we exclusively report the transformer-based methods [36, 39] performance on the real-world NYUD-v2 benchmark.

Table 11. **Full quantitative comparisons on our proposed photorealistic *OB-Hypersim*.**

Method	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
Depth Baseline [41]	0.6948	0.3739	0.3123	0.2481	0.1342	0.4759	-	-
OB Baseline (ours)	-	-	-	-	-	-	0.8099	**0.5734**
SharpNet [17]	0.8551	0.8120	0.4422	0.4037	0.2498	0.3561	0.7342	0.4743
MTAN [12]	0.8050	0.4703	0.3765	0.3323	0.1660	0.4023	0.7430	0.3575
PAD-Net [30]	0.8404	0.4748	0.4322	0.3949	0.1688	0.3866	0.6732	0.3555
MTI-Net [24]	0.7560	0.4194	0.3746	0.3183	0.1510	0.4365	0.7490	0.3976
InvPT [38]	0.9018	0.5154	0.5106	0.4778	0.1858	0.3555	0.8004	0.3409
DenseMTL [13]	0.7475	0.4141	0.4095	0.3485	0.1465	0.4410	0.8520	0.4849
Ours	0.6583	0.3463	0.2963	0.2279	**0.1223**	**0.5167**	0.8670	0.5163
Ours + *SSR*	**0.6537**	**0.3456**	**0.2954**	**0.2266**	0.1243	0.5148	**0.8732**	0.5109

We employ the NYUv2-OC++ dataset [16, 18], a manually annotated version of the NYUD-v2 test set with occlusion boundaries, to evaluate joint depth and OB estimation in more real-world scenarios. While these OB annotations (a specialized subset of OBs) deviate from the canonical definition in [26], they yield practical training benefits. Our evaluation protocol trains on NYUv2-OC++ (using both depth and OB ground truth) while testing depth estimation performance on the original NYUD-v2 training set, with quantitative results shown in Table 12.

Table 12. Quantitative comparisons using NYUv2-OC++ as the training set and evaluating on the NYUD-v2 test set.

Method	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑
Depth Baseline [41]	0.4213	0.1436	0.1129	0.0667	0.0492	0.8754
MTAN [12]	0.5558	0.1927	0.1575	0.1211	0.0668	0.7751
PAD-Net [30]	0.6583	0.2346	0.1980	0.1750	0.0811	0.6926
MTI-Net [24]	0.5598	0.1951	0.1627	0.1255	0.0674	0.7673
InvPT [38]	0.5069	0.1734	0.1415	0.1022	0.0594	0.8176
DenseMTL [13]	0.5166	0.1812	0.1508	0.1103	0.0624	0.8011
MLoRE [36]	0.4751	0.1588	0.1313	0.0880	0.0558	0.8372
Ours	0.4055	0.1385	0.1114	0.0638	0.0475	0.8837
Ours *SSR*	**0.4035**	**0.1380**	**0.1097**	**0.0631**	**0.0472**	**0.8857**

In Table 13, we present the official results of additional MT models reported in their respective papers, alongside our model trained on NYUD-v2. We re-evaluated *MoDOT* performance using the MT learning evaluation protocol adopted from MTI-Net [24], which differs from the evaluation protocol used in NeWCRFs [41] in the main paper. While our method achieves the best depth prediction performance, its OB performance is comparatively weaker. This can be attributed to several key factors:

- This variant of our model was initially selected for its superior performance on the synthetic OB-FUTURE dataset, which contains complete and accurate OB annotations. When applied to NYUD-v2 with its pseudo and imperfect OB labels, we maintained the original model configuration-adjusting only the loss weights, without further architectural modifications or network structure innovations.
- Prior MT learning methods trained on NYUD-v2 typically optimize across four tasks—depth estimation, object boundary detection, surface normal estimation, and semantic segmentation. This joint training strategy has been shown to improve overall performance across all tasks [38, 39]. In contrast, our method focuses only on two tasks: depth estimation and object boundary (pseudo OB) estimation, which may limit the auxiliary geometry benefits that arise from learning with more related tasks.
- The evaluation protocol and model selection criteria used in our experiments and ablation studies differ from the standard MATLAB-based evaluation (reasons were explained in the main paper). Specifically, we evaluated our method using a fixed F-score threshold (*i.e.*, 0.7, as reported in the main paper), whereas MATLAB-based evaluations typically report the best F-score across a wide range of thresholds (from 0.01 to 0.99) and allow a tolerance distance between predicted and ground-truth edges (in Table 8, we find that these two evaluation metrics are not linearly correlated). Our model has been specifically selected and tuned for this fixed-threshold setup, which may affect the comparability of results under MATLAB-based one.

Table 13. Official Comparison on NYUD-v2 using the best MT learning results reported in their respective papers. Unreported parameters (Params) and GFLOPs are indicated with a dash -.

Method	**Backbone**	**Params (M)**	**GFLOPs (G)**	**Depth RMSE ↓**	**Boundary odsF ↑**
Cross-Stich [14]	HRNet18	-	-	0.6290	76.38
PAD-Net [30]	HRNet18	81	124	0.6270	76.38
PAD [42]	HRNet18	-	-	0.6178	76.42
PSD [43]	HRNet18	-	-	0.6246	76.42
ATRC [2]	ResNet50	96	216	0.5363	77.94
MTI-Net [24]	HRNet18	128	161	0.5365	77.86
InvPT [38]	ViT-L	423	669	0.5183	78.10
DenseMTL [13]	ResNet101	-	-	0.5930	-
TaskPrompter [39]	ViT-L	401	497	0.5152	78.20
TaskExpert [39]	ViT-L	420	622	0.5157	78.40
MLoRE [36]	ViT-L	571	407	0.5076	78.43
DeMT [33]	Swin-S	53.03	121.05	0.5474	78.10
MQTransformer [32]	Swin-L	204.3	365.25	0.5325	78.20
SEM [6]	VIT-L	-	-	0.4937	78.40
MTMamba [10]	Swin-L	307.99	540.81	0.5066	**78.70**
InvPT ++ [40]	VIT-L	402	-	0.5096	78.10
TSP-Transformer [29]	Vit-L	402.34	1146.24	0.4961	77.50
TaskDiffusion [37]	-	-	-	0.5020	78.64
TaskDiffusion [37] /w MLoRE	-	-	-	0.5033	78.89
InvPT + DTME-MTL [7]	ViT-L	-	-	0.5020	78.20
Taskprompter + DTME-MTL [7]	ViT-L	-	-	0.5122	78.40
Our	Swin-L	281.01	385.86	0.4845	58.9
Our + *SSR*	Swin-L	281.42	385.89	**0.4830**	59.3

C.5. Additional Ablation Study

The ablation study in this section focuses on the stage one model in *MoDOT* trained on OB-FUTURE, which provides more accurate OB ground truth. In Table 14, we present an ablation study of the encoder architecture used in *MoDOT*, showing consistent performance improvements as the Swin Transformer backbone capacity increases from Tiny to Large.

Table 14. **Ablation on the shared encoder backbone (all pretrained with a window size of 7).**

Details	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
Swin-Tiny	0.4832	0.1238	0.1089	0.0750	0.0461	0.8988	0.9175	0.5884
Swin-Small	0.4659	0.1209	0.1091	0.0725	0.0451	0.9042	0.9195	0.5821
Swin-Base	0.4377	0.1125	0.0992	0.0617	0.0418	0.9042	**0.9259**	0.5761
Swin-Large	**0.3963**	**0.1020**	**0.0901**	**0.0523**	**0.0380**	**0.9427**	0.9090	**0.6131**

In Table 15, we present an ablation study on several representative loss weight settings. Due to the large number of possible combinations, we explored only a limited subset of examples. The results suggest that even better performance may be achievable with more exhaustive tuning. Specifically, the first row corresponds to training without side-output supervision, the second row shows typical multi-task loss weights for edge detection, the fifth row lists loss weight combinations used for NYUD-v2, and the last row in the first block presents settings for two synthetic datasets. Additionally, in the final block, we evaluate *OBDCL* under weaker and stronger weighting schemes. A weight of 0.1 yields the best performance among these three, indicating that the loss is sensitive to its scaling, as previously discussed.

Table 15. **Ablation on OB, depth, and OBDCL loss weights.** Varying the weights reveals their impact on performance.

Depth Weight	**OB Weight**	**OBDCL Weight**	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall↑	OB-Fscore↑
1.0	0.0, 0.0, 0.0, 0.0, 1.0	0.1	0.4009	0.1026	0.0895	0.0509	0.0380	**0.9452**	0.9100	0.6017
1.0	0.0, 0.0, 0.0, 0.0, 50.0	0.1	0.4070	0.1047	0.0920	0.0540	0.0386	0.9412	0.9305	0.6147
1.0	1.0, 1.0, 1.0, 1.0, 1.0	0.1	0.4075	0.1049	0.0928	0.0542	0.0389	0.9415	0.9311	0.5721
1.0	0.5, 0.5, 0.5, 0.5, 1.0	0.1	0.4068	0.1043	0.0921	0.0536	0.0386	0.9436	0.9105	0.6051
1.2	0.5, 1.5, 1.5, 2.5, 5.0	0.1	0.3978	**0.1017**	**0.0891**	**0.0500**	**0.0376**	0.9436	0.9123	**0.6205**
1.2	0.1, 0.3, 0.3, 0.5, 2.3	0.1	**0.3963**	0.1020	0.0901	0.0523	0.0380	0.9427	0.9090	0.6131
1.2	0.1, 0.3, 0.3, 0.5, 2.3	0.01	0.4144	0.1069	0.0961	0.0569	0.0399	0.9356	0.9107	0.6147
1.2	0.1, 0.3, 0.3, 0.5, 2.3	1.0	0.4082	0.1073	0.0912	0.0536	0.0384	0.9439	**0.9327**	0.5729

To demonstrate the simplicity and effectiveness of our *CASM* design, we conducted an ablation study (Table 16) by systematically removing internal components or adding supplementary ones. The fourth row shows results without parallel enhanced depth computation, using cross-channel attention enhanced depth/OB features instead of upsampled features as inputs for MSS-Fuse.

Table 16. Ablation study of internal components in *CASM* (CA: Channel Attention, SA: Spatial Attention).

Details	RMSE↓	$RMSE_{log}$ ↓	Abs Rel↓	Sq Rel↓	log10↓	$\delta < 1.25$ ↑	OB-Recall ↑	OB-Fscore ↑
Remove two cross CA	0.4140	0.1050	0.0907	0.0530	0.0386	0.9435	0.9231	0.5998
Remove Depth CA	0.4023	0.1027	0.0901	0.0531	0.0380	0.9439	0.9331	0.5795
Remove OB CA	0.4143	0.1055	0.0912	0.0539	0.0389	0.9409	**0.9343**	0.5754
Series pass cross CA then MSS-Fuse	0.4017	0.1032	**0.0892**	0.0512	0.0380	0.9436	0.9295	0.5819
Remove MSS-Fuse	0.4109	0.1060	0.0926	0.0546	0.0391	0.9386	0.9257	0.5863
Remove local 3×3 convolotions in MSS-Fuse	0.4163	0.1067	0.0952	0.0562	0.0397	0.9380	0.9234	0.5892
Add additional Depth SA for OB	0.3989	**0.1020**	**0.0892**	**0.0506**	**0.0378**	**0.9446**	0.9186	0.5998
Final *CASM*	**0.3963**	**0.1020**	0.0901	0.0523	0.0380	0.9427	0.9090	**0.6131**

D. More Visualization Results

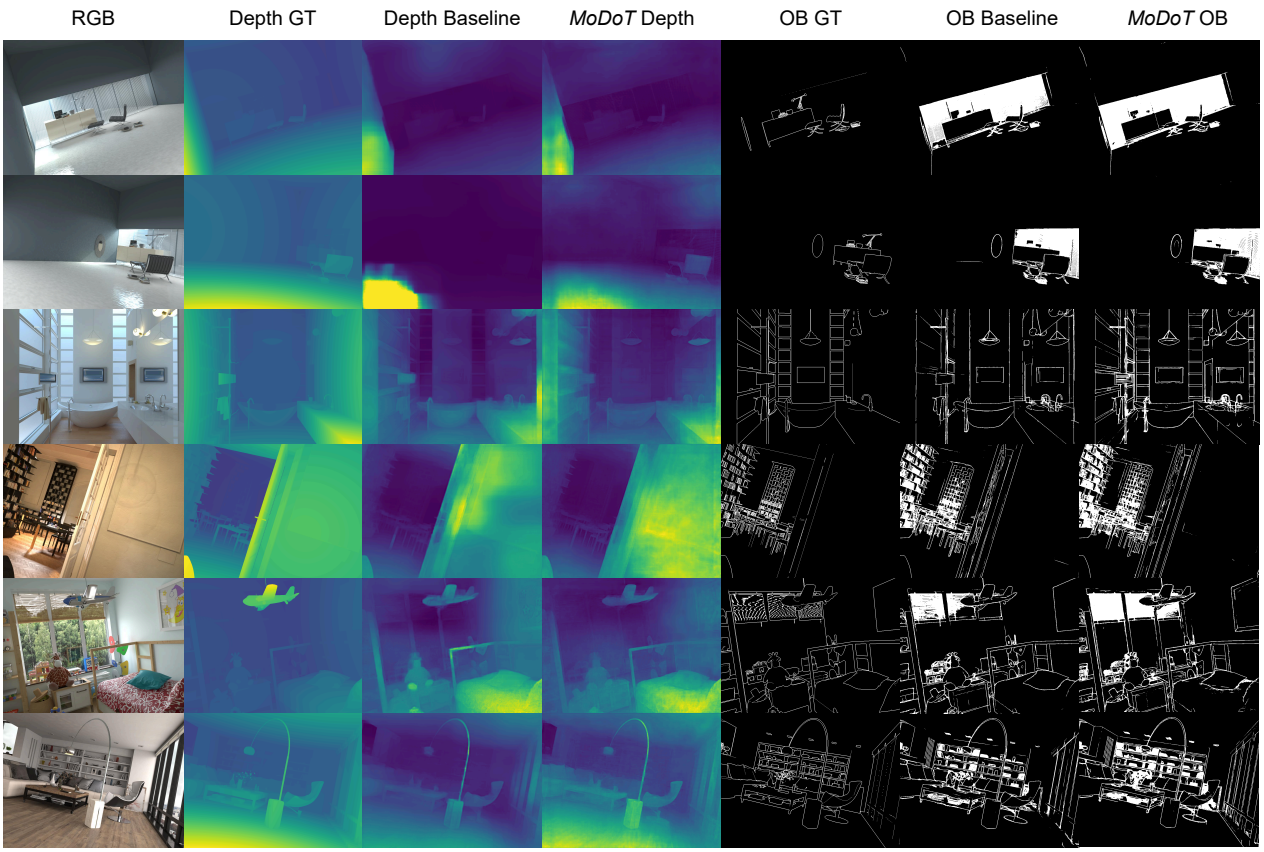

Figure 7. Additional qualitative comparisons on our photorealistic *OB-Hypersim*.

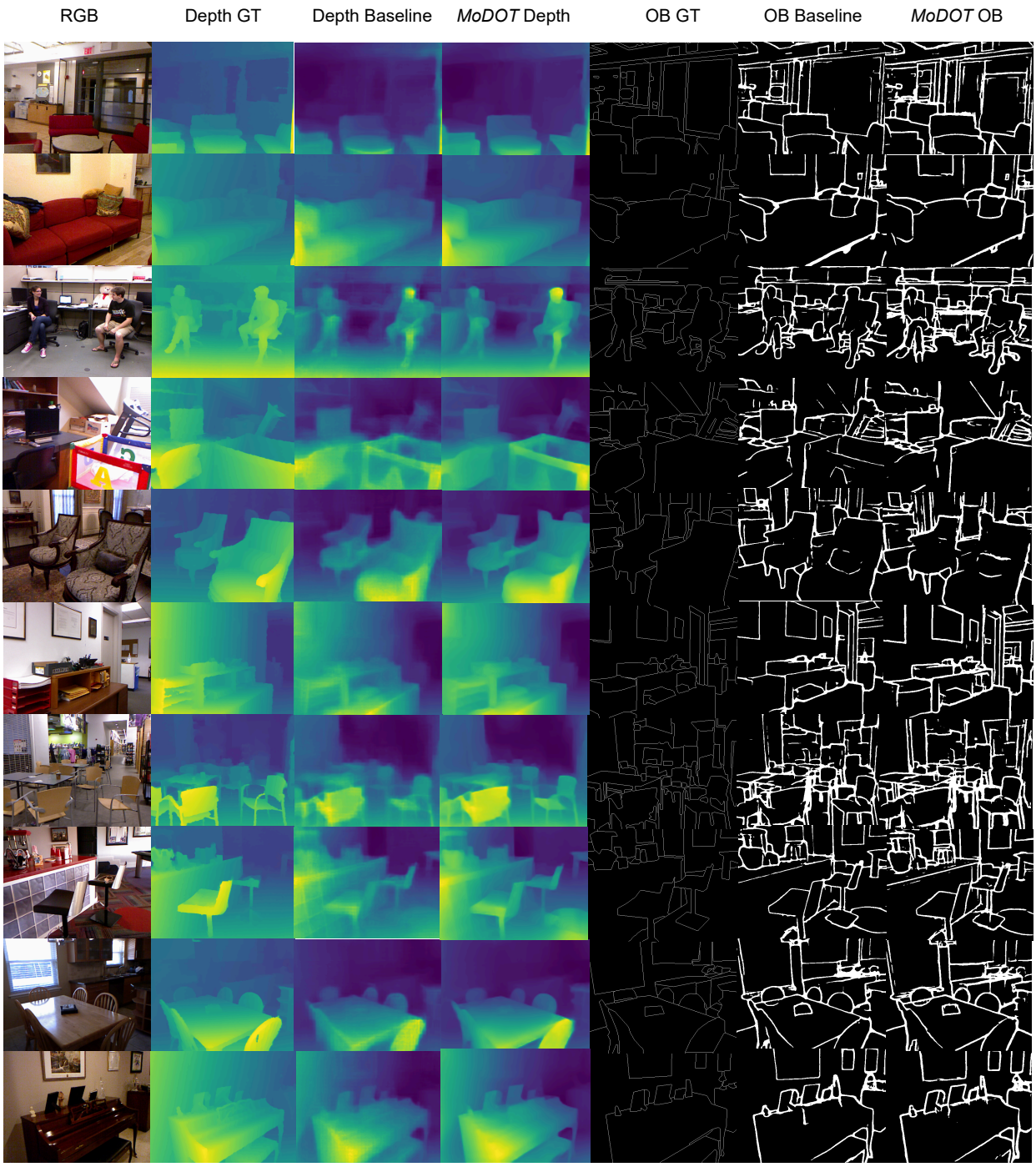

Figure 8. Additional qualitative comparisons on the real-world NYUD-v2.

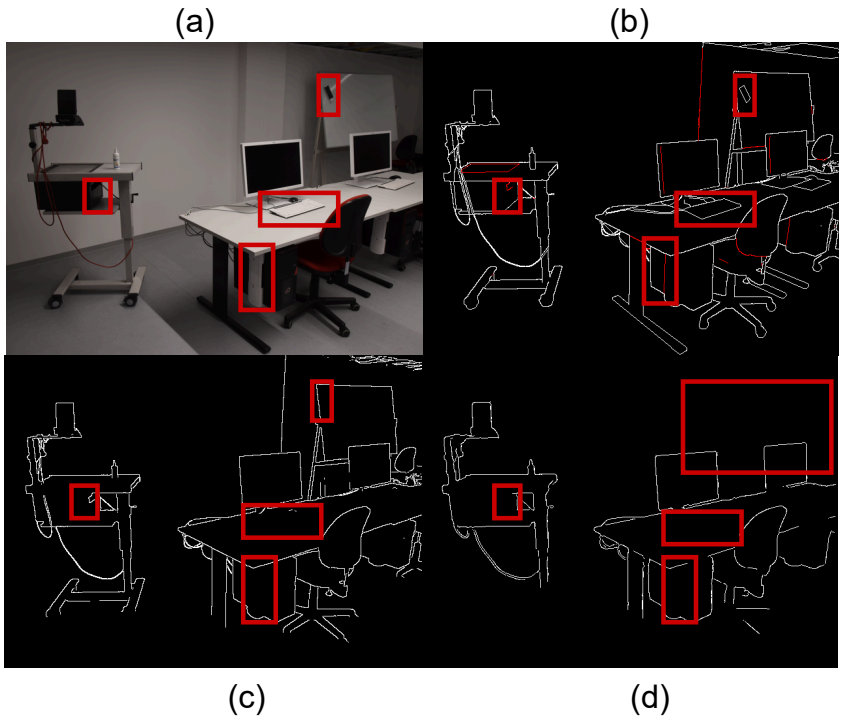

Figure 9. Zoom-in view of OBs in Figure 1 of the main paper. (a) An RGB image from iBims-1 [9]. (b) Our annotated/targeted OBs. (c) Incomplete OBs directly generated by the method in P2ORM [16]. (d) Depth edges provided in the dataset. Our targeted OB annotations provide richer geometric details and comprehensive scene structures.

References

[1] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5861–5870, 2023. 3

[2] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15869–15878, 2021. 11

[3] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems (NIPS)*, 27, 2014. 1

[4] Panhe Feng, Qi She, Lei Zhu, Jiaxin Li, Lin Zhang, Zijian Feng, Changhu Wang, Chunpeng Li, Xuejing Kang, and Anlong Ming. Mt-orl: Multi-task occlusion relationship learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9364–9373, 2021. 1, 2, 8

[5] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision (IJCV)*, 129:3313–3337, 2021. 2

[6] Huimin Huang, Yawen Huang, Lanfen Lin, Ruofeng Tong, Yen-Wei Chen, Hao Zheng, Yuexiang Li, and Yefeng Zheng. Going beyond multi-task dense prediction with synergy embedding models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28181–28190, 2024. 11

[7] Wooseong Jeong and Kuk-Jin Yoon. Resolving token-space gradient conflicts: Token space manipulation for transformer-based multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2887–2897, 2025. 11

[8] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[9] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 0–0, 2018. 15

[10] Baijiong Lin, Weisen Jiang, Pengguang Chen, Yu Zhang, Shu Liu, and Ying-Cong Chen. MTMamba: Enhancing multi-task dense scene understanding by mamba-based decoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 11

[11] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22290–22300, 2023. 8

[12] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 1871–1880, 2019. 8, 9, 10

[13] Ivan Lopes, Tuan-Hung Vu, and Raoul de Charette. Cross-task attention mechanism for dense multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2329–2338, 2023. 8, 9, 10, 11

[14] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016. 11

[15] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High quality entity segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4024–4033, 2023. 2

[16] Xuchong Qiu, Yang Xiao, Chaohui Wang, and Renaud Marlet. Pixel-pair occlusion relationship map (p2orm): formulation, inference and application. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–708, 2020. 3, 5, 8, 10, 15

[17] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (CVPRW)*, pages 0–0, 2019. 5, 8, 10

[18] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14648–14657, 2020. 2, 5, 10

[19] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021. 3, 6

[20] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in Neural Information Processing Systems (NIPS)*, 18, 2005. 9

[21] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760, 2012. 2

[22] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760, 2012. 2, 5

[23] Xavier Soria, Angel Sappa, Patricio Humanante, and Arash Akbarinia. Dense extreme inception network for edge detection. *Pattern Recognition*, 139:109461, 2023. 8

[24] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 527–543, 2020. 2, 8, 9, 10, 11

[25] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 2, 9

[26] Chaohui Wang, Huan Fu, Dacheng Tao, and Michael J Black. Occlusion boundary: A formal definition & its detection via deep exploration of context. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44:2641–2656, 2020. 2, 10

[27] Guoxia Wang, Xiaochuan Wang, Frederick WB Li, and Xiaohui Liang. Doobnet: Deep object occlusion boundary detection from an image. In *Asian Conference on Computer Vision (ACCV)*, pages 686–702, 2019. 8

[28] Peng Wang and Alan Yuille. Doc: Deep occlusion estimation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 545–561, 2016. 2

[29] Shuo Wang, Jing Li, Zibo Zhao, Dongze Lian, Binbin Huang, Xiaomei Wang, Zhengxin Li, and Shenghua Gao. Tsp-transformer: Task-specific prompts boosted transformer for holistic scene understanding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 925–934, 2024. 11

[30] 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 (CVPR)*, pages 675–684, 2018. 8, 9, 10, 11

[31] Lintao Xu and Chaohui Wang. Interactive occlusion boundary estimation through exploitation of synthetic data. *British Machine Vision Conference (BMVC)*, 2025. 2, 3, 5, 8

[32] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 34:1228–1240, 2023. 11

[33] Yangyang Xu, Yibo Yang, and Lefei Zhang. Demt: Deformable mixer transformer for multi-task learning of dense prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3072–3080, 2023. 11

[34] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381, 2024. 5, 9

[35] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems (NIPS)*, pages 21875–21911, 2024. 3, 5

[36] Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. Multi-task dense prediction via mixture of low-rank experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 8, 10, 11

[37] Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. Multi-task dense predictions via unleashing the power of diffusion. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. 11

[38] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–530. Springer, 2022. 2, 8, 9, 10, 11

[39] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 8, 10, 11

[40] Hanrong Ye and Dan Xu. Invpt++: Inverted pyramid multi-task transformer for visual scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):7493–7508, 2024. 11

[41] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3916–3925, 2022. 1, 2, 3, 9, 10, 11

[42] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4106–4115, 2019. 11

[43] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4514–4523, 2020. 11