Replica
FRL-Apartment 2
FRL-Apartment 4
HM3D
house spaces
business spaces
Hypersim
near-range
medium-range
far-range
VA
NYUv2

# Supplementary Materials

## A Overview

- In Sec. B we provide examples of each dataset we adopt.
- In Sec. C we provide more explanations and analysis to depth-relevant and depth-irrelevant features.
- In Sec. D we extend the discussion in main paper Sec. 4.
- In Sec. E we show the formula of depth evaluation metrics and organize TL;DR for terms used in the work.
- In Sec. F we display the pseudo-code for direct supervised learning and meta-learning under our fine-grained task setting.
- In Sec. G we provide more studies on different learning strategies, compare with other cross-dataset evaluation works, plug meta-initialization into existing frameworks to validate meta-learning, show extensive qualitative depth map results, and show more quantitative and qualitative results for depth-supervised NeRF.
- In Sec. H we illustrate the broader impact related to this work.

## B Data Samples

Examples of all the adopted datasets and their features are shown in Fig. S1.

## C Depth-Relevant and Depth-Irrelevant Features

In Introduction and Section 3.1 of the main paper, we explain the division between depth-relevant and depth-irrelevant features: whether pixel color or appearance changes indicate depth changes. An example of the former is foreground object boundaries, where the color changes imply depth changes. By contrast, simple material textures or paintings are depth-irrelevant. We show an illustration in Fig. S2. In Fig. 3 of the paper, we show that meta-learning can induce better image-to-depth understanding and suppress depth-irrelevant features, such as flat areas on the textured carpet and clearer object boundaries in depth maps. For deeper insight, the experiment in paper Fig. 3 is trained on Replica, which contains only 18 environments, and some possess the same structures with minor arrangement changes (See Fig. S1). Limited scene variety makes it difficult for direct supervised learning to attain good or valid image-to-depth understanding, and thus it reaches inferior performance. Therefore, as shown in paper Fig. 3, it cannot suppress depth-irrelevant cues and reflect texture patterns in the depth maps.

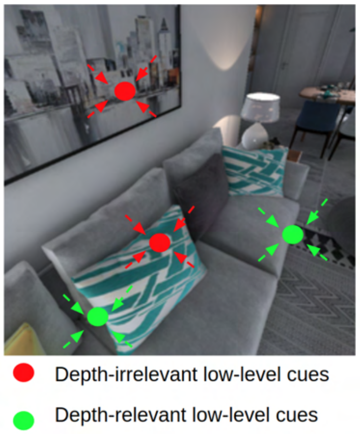

🔴 Depth-irrelevant low-level cues
🟢 Depth-relevant low-level cues

Figure S2: **Illustration of the division between depth-relevant and depth-irrelevant cues.**

In contrast, meta-learning is good at few-shot or low-source learning in literature because of its learning nature. It can learn generalizable good or valid image-to-depth mappings from scenes with limited scene variety. Its dual-step optimization does not directly fit each training pair but uses a step size $\beta$ in Algorithm 1 (meta-initialization) to control how much the explored gradient updating direction is trusted. Thus, it avoids directly fitting each seen example and allows more exploration in the neighborhood of each solution point, achieving better image-to-depth understanding for higher depth accuracy. We show a comparison in Fig. S3 for the effects of the $\beta$ parameter.

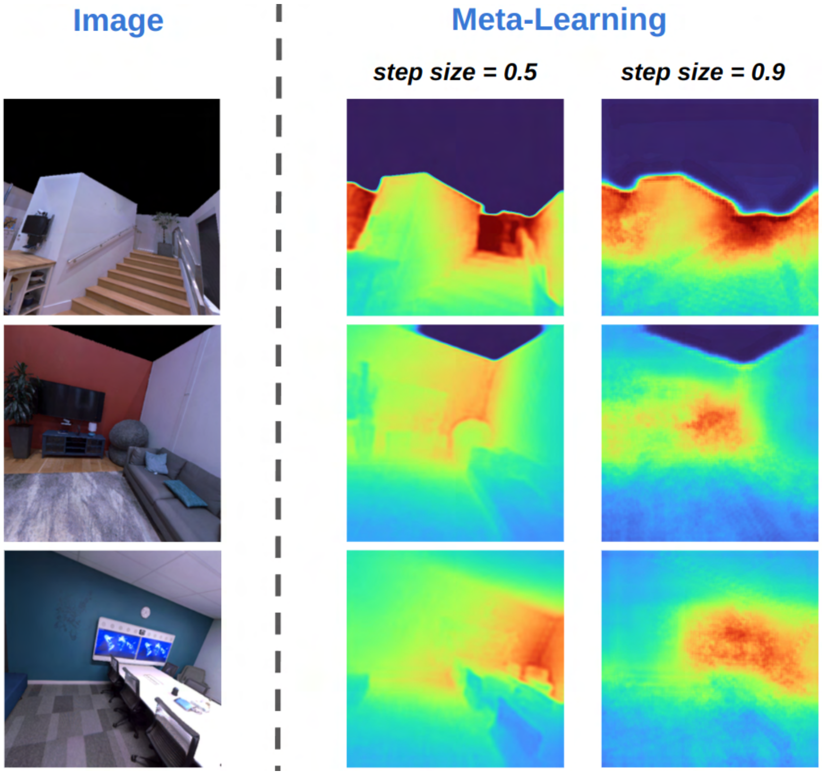


Figure S3: **Effects of different step size parameter $\beta$.** ConvNeXt-base architecture is used. We show that using a larger step size parameter $\beta = 0.9$, the training becomes more similar to direct supervised learning that tries to fit each seen training sample, but may not fully explore the neighborhood for each intermediate solution and attain better image-to-depth mappings as meta-learning performs. See the main text.

## D How is fine-grained task related to other meta-learning studies?

There are several previous findings on learning techniques or issues related to meta-learning. Here we discuss how those findings apply to fine-grained tasks.

**Relation to domain-agnostic task augmentation**. Domain-agnostic task augmentation is to densify sampled data points in each task to add robustness, such as label noise

**Relation to task interpolation**. Unlike domain-agnostic augmentation, task interpolation (MLTI

**Relation to meta-memorization and meta-overfitting**. Prior meta-learning studies

## E Error Metrics Formula for Depth Evaluation and Term Dictionary

We provide formula for adopted depth evaluation metrics between prediction ($x_s$) and groundtruth ($y_s$), $\forall s \in S$, as follows.

(1) MAE: $\frac{1}{|S|}\sum_{s\in S}|x_s - y_s|$.

(2) AbsRel: $\frac{1}{|S|}\sum_{s\in S}\frac{|x_s - y_s|}{y_s}$.

(3) RMSE: $\sqrt{\frac{1}{|S|}\sum_{s\in S}(x_i - y_i)^2}$.

(4) RMSE$_{log}$: $\sqrt{\frac{1}{|S|}\sum_{s \in S}(log(x_i) - log(y_i))^2}$.

The above four are error metrics. The lower the better.

(5) Depth accuracy $\delta_i$:

$$\delta_i = \frac{\text{card}(\left\{x_s : \max\{\frac{x_s}{y_s}, \frac{y_s}{x_s}\} < 1.25^i\right\})}{\text{card}(\{y_s\})}, \tag{S1}$$

where card(.) is the cardinality of a set. This is an accuracy metric. The higher the better.

**Variance**. We also observe that meta-initialization induces smaller variances on error $|x_s - y_s|$. Error variance for Table 4 Hypersim→Replica are *DPT-hybrid*: 0.236, *DPT-hybrid + Meta*: 0.201, *DPT-large*: 0.218, *DPT-large + Meta*: 0.199. This shows that meta-initialization tends to predict more structured depth in reasonable ranges, preventing jumpy depth that causes large errors.

**Dictionary**. We provide quick explanations as TL;DR for terms used in the paper.

- **generalizability** refers to whether a pretrained model can generalize to unseen data and make reasonably good inferences. This work especially stresses generalizability to unseen data from different datasets.
- **zero-shot cross-dataset inference** refers to training on $A$-dataset without any knowledge on $B$-dataset and making inference on $B-$dataset.
- **scene variety** refers to variety of scene appearance and geometry (RGBD) pairs in a dataset.
- **task** in meta-learning context contains a distribution to sample data from. Those data share similarities or affinity so that they can be grouped together.
- **depth-relevant/ depth-irrelevant low-level cues** refer to whether pixel color or appearance changes as low-level cues indicate depth changes. An example of the former is foreground object boundaries. Simple material textures or paintings are depth-irrelevant.

## F Pseudo-code

We display pseudo-code for direct supervised learning and our fine-grained task meta-learning as follows. Our fine-grained task meta-learning only needs to adapt a few lines of codes in a conventional supervised learning framework to build bi-level optimization. With this simple plug-in, our fine-grained task meta-learning effectively learns better domain generalizability and higher geometry resolvability, as shown in the qualitative and quantitative evaluation.

```
# I: image as a minibatch
# D: depth groundtruth of I
def direct_supervised_learning(I,D):
    optimizer.zero_grad() # flush out gradient
    D_pred = model(I) # predict depth
    loss = criterion(D_pred,D) # calculate loss
    loss.backward() # back-prop
    optimizer.step() # update network

def meta_learning(I,D):
    meta_optimizer.zero_grad() # flush out gradient
    for step in range(L): # L-step
        inner_optimizer.zero_grad() # flush out gradient
        D_pred = inner_model(I) # predict depth
        loss = criterion(D_pred,D) # calculate loss
        loss.backward() # back-prop
        optimizer.step() # SGD-update inner-network
    for meta_param, inner_param = zip(meta_model.parameters(),
    inner_model.parameters()):
        # assign gradient as parameter difference
        meta_param.grad = meta_param - inner_param
    meta_optimizer.step() # SGD-update meta-network
```

Listing 1: *PyTorch-like* pseudo-code for direct supervised learning and our fine-grained task meta-learning

## G More Results

**Comparison with simple pretraining on the same dataset**. We first compare with *simple pretraining* on the same dataset but with different learning schedules. The networks are pretrained by 5 epochs using larger and feasible learning rates of 0.001 and 0.0003 and a strong weight decay of 0.1. Then, the learned weights serve as initialization for the following supervised learning, whose setting is the same as in the main paper. The purpose is trying to examine whether a higher-level and smooth prior can be learned without using meta-learning. Then the same as meta-initialization, we use the learned weights as initialization for the second-stage supervised learning. We use ResNet50 and ConvNext-base and train/test on NYUv2. Results of different learning rates and weight decay are compared and shown in Table S1. We find that larger learning rates and weight decay cannot learn a good prior but damage the performance. Besides, higher weight decay did not result in apparent positive effects. Note that "w/o Pretraining" simply uses the second-stage supervised learning. The entry "w/ Pretraining (lr=3x$10^{-4}$, wd=$10^{-2}$)" is equivalent with longer training for "w/o Pretraining" since its learning rate and weight decay match those used in the "w/o Pretraining." The results show that simple pretraining cannot learn a better prior. Thus, we resort to meta-learning with its advantages of higher model generalizability in literature.

Table S1: **Comparison with simple pretraining strategy.** Adopted architecture, learning rate (lr), and weight decay (wd) are shown. The pretraining first uses a higher lr and wd of 0.001 and 0.1 to learn a smooth prior. We also experiment with different lr and wd for comparison. The learned weights are then used as initialization for the second-stage supervised learning. See text for the pretraining setting. The pretraining does not improve over baseline without this trick, and larger weight decay slightly degrades the performance.

| NYUv2 | MAE | AbsRel | RMSE | $\delta_1$ | $\delta_2$ | $\delta_3$ |
|---|---|---|---|---|---|---|
| ResNet50 | | | | | | |
| w/o Pretraining | 0.345 | 0.131 | 0.480 | 83.6 | 96.4 | 99.0 |
| w/ Pretraining (lr=$10^{-3}$, wd=$10^{-1}$) | 0.362 | 0.138 | 0.500 | 82.9 | 95.6 | 97.9 |
| w/ Pretraining (lr=3x$10^{-4}$, wd=$10^{-1}$) | 0.347 | 0.132 | 0.481 | 83.5 | 96.4 | 99.0 |
| w/ Pretraining (lr=3x$10^{-4}$, wd=$10^{-2}$) | 0.345 | 0.133 | 0.480 | 83.6 | 96.4 | 99.0 |
| w/ Meta-Initialization | **0.325** | **0.122** | **0.454** | **85.4** | **96.8** | **99.3** |
| ConvNeXt-base | | | | | | |
| w/o Pretraining | 0.273 | 0.101 | 0.394 | 89.4 | 97.9 | **99.5** |
| w/ Pretraining (lr=$10^{-3}$, wd=$10^{-1}$) | 0.288 | 0.109 | 0.414 | 87.5 | 97.5 | 99.4 |
| w/ Pretraining (lr=3x$10^{-4}$, wd=$10^{-1}$) | 0.276 | 0.103 | 0.397 | 89.2 | 97.9 | **99.5** |
| w/ Pretraining (lr=3x$10^{-4}$, wd=$10^{-2}$) | 0.274 | 0.101 | 0.395 | 89.3 | 97.9 | **99.5** |
| w/ Meta-Initialization | **0.266** | **0.099** | **0.387** | **89.8** | **98.1** | **99.5** |

**Comparison with gradient accumulation**. We next compare with gradient accumulation. Setting-1: Similar to the prior-learning stage, gradients are accumulated for 4 iterations and then used to update network parameters once. This resembles taking off the inner exploration and inner optimizer in meta-learning. We train this strategy for 5 epochs with a learning rate of 0.0012, 4x by its base learning rate since we accumulate gradients for 4 iterations. Then we used the learned weights as initialization for the following standard supervised learning whose hyperparameters are the same as in the main paper. Setting-2: We adopt a single-stage approach, which does not require the prior-learning stage, and simply use the gradient accumulation trick in the standard supervised learning. We also accumulate gradients for 4 iterations and use a learning rate of 0.0012. The rest hyperparameters are intact. We again use ResNet50 and ConvNext-base and train/test on NYUv2. Results are shown in Table S2. From the table, we empirically find both settings do not improve the results of "Base", which is standard supervised learning without any add-on methods. We think this is because gradient accumulation has effects of using large batch size, which has a higher risk to overfit training data by converging to poor local optima, due to the reduction of stochasticity in the gradient updates

**Additional comparison to other works**. Few recent works are related to cross-dataset evaluation for indoor depth, including unsupervised domain adaptation

We further compare to an unsupervised domain adaptation method T$^2$Net

**Additional results for adding meta-initialization to dedicated depth estimation architecture**. In addition to Table 4 in the paper, we provide more experiments using other existing dedicated architecture for monocular depth estimation, including AdaBins

Table S2: **Comparison with gradient accumulation.** Two settings in comparison are described in Sec. G. "Base" refers to using standard supervised learning without any add-on methods. Empirically we find gradient accumulation does not improve results but degrades performance a little.

| NYUv2 | MAE | AbsRel | RMSE | $\delta_1$ | $\delta_2$ | $\delta_3$ |
|---|---|---|---|---|---|---|
| ResNet50 | | | | | | |
| Base | 0.345 | 0.131 | 0.480 | 83.6 | 96.4 | 99.0 |
| Setting-1 | 0.349 | 0.133 | 0.487 | 83.4 | 96.3 | 99.0 |
| Setting-2 | 0.353 | 0.134 | 0.493 | 83.1 | 96.1 | 98.8 |
| Meta-Initialization | **0.325** | **0.122** | **0.454** | **85.4** | **96.8** | **99.3** |
| ConvNeXt-base | | | | | | |
| Base | 0.273 | 0.101 | 0.394 | 89.4 | 97.9 | **99.5** |
| Setting-1 | 0.277 | 0.103 | 0.399 | 89.1 | 97.8 | 99.4 |
| Setting-2 | 0.279 | 0.105 | 0.406 | 88.9 | 97.7 | 99.4 |
| Meta-Initialization | **0.266** | **0.099** | **0.387** | **89.8** | **98.1** | **99.5** |

| SUNCG→NYUv2 | AbsRel | RMSE | $\delta_1$ | $\delta_2$ | $\delta_3$ |
|---|---|---|---|---|---|
| | 0.203 | 0.738 | 67.0 | 89.1 | 96.6 |
| | 0.186 | 0.710 | 71.2 | 91.7 | 97.7 |
| | 0.196 | 0.662 | 69.5 | 91.0 | 97.2 |
| Our Meta-Initialization | **0.177** | **0.635** | **72.8** | **92.8** | **97.8** |

**Qualitative results on zero-shot cross-dataset evaluation.** Following the quantitative comparison in Table 3 and Table 4 in the main paper, we show qualitative comparisons to examine zero-shot cross-dataset evaluation. In Fig. S4, we use ConvNeXt-Base as the backbone network, train on HM3D and make inferences on Replica and VA, and compare between using meta-initialization and without meta-initialization. In Fig. S5, we use the dedicated depth estimation architecture, DPT-large, train on Hypersim and make inferences on NYUv2 and Replica, and also compare between using meta-initialization and without meta-initialization.

In both Fig. S4 and S5, meta-initialization induces clearer depth shapes and outlines with less irregularity. The results show that on the challenging zero-shot cross-dataset evaluation, meta-initialization can learn higher model generalizability that transfers knowledge from synthetic datasets (HM3D and Hypersim) to more challenging and higher quality synthetic (VA) or real data (NYUv2) and estimates accurate depth shapes for them.

**Depth-supervised NeRF**. In the main paper Sec. 4.4 we train NeRF [1] with supervision by depth predicted from our meta-initialization strategy using ConvNeXt-Base backbone. To convert from depth (z-value) to ray distance in a pinhole camera model, we do the following conversion.

$$distance = depth \times \sqrt{1 + (\frac{x - c_x}{f_x})^2 + (\frac{y - c_y}{f_y})^2}, \tag{S2}$$

where $c_x$ and $c_y$ are principal point coordinates, and $f_x$ and $f_y$ are focal lengths.

We show more quantitative comparisons for depth-supervised NeRF in Table S5 on Replica. Each is trained with 180 views along with losses for pixel color and distances, as described in the main paper Sec.4.4. The use of meta-initialization consistently outperforms the baselines, without meta-initialization, in terms of image quality metrics. More qualitative comparisons are displayed in Fig. S6.

## H Broader Impact

The research focuses on using gradient-based meta-learning to improve monocular depth estimation performance. As monocular depth can be applied in indoor AR/VR creation and interaction, robot navigation, and learning 3D representations for general purposes, the proposed method can be a part of a training convention that facilitates depth estimation to attain each goal for each application, especially fulfill the purpose of in-the-wild robustness.

**Ethical considerations**: This work studies how to improve model generalizability by meta-learning. The advantage this work brings about is better indoor depth estimation technology for applications

[1] Specifically, we use high-performing instant-ngp

Table S4: **Extended comparison of zero-shot cross-dataset evaluation for dedicated depth estimation architecture.**

| Hypersim → Replica | MAE | AbsRel | RMSE | $\delta_1$ | $\delta_2$ | $\delta_3$ |
|---|---|---|---|---|---|---|
| | 0.395 | 0.210 | 0.564 | 70.4 | 88.1 | 94.7 |
| AdaBins+Meta | **0.377** | **0.198** | **0.541** | **71.6** | **89.2** | **95.5** |
| | 0.352 | 0.191 | 0.522 | 73.0 | 90.9 | 96.5 |
| GLPDepth+Meta | **0.337** | **0.180** | **0.498** | **74.4** | **92.2** | **96.8** |
| Hypersim → NYUv2 | MAE | AbsRel | RMSE | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| | 0.469 | 0.188 | 0.642 | 72.6 | 91.2 | 96.6 |
| AdaBins+Meta | **0.448** | **0.175** | **0.625** | **74.0** | **92.6** | **97.4** |
| | 0.438 | 0.169 | 0.604 | 75.3 | 93.9 | 98.2 |
| GLPDepth+Meta | **0.414** | **0.158** | **0.583** | **77.9** | **94.3** | **98.3** |

Table S5: **More results on depth-supervised NeRF.** We test on Replica 'room-0', 'room-1', room-2', 'office-0', 'office-1', and 'office-2' environments. We train a NeRF on each environment with 180 views. The comparison between using depth from meta-initialization and w/o meta-initialization for supervision is drawn. PSNR and SSIM are image quality metrics; the higher, the better.

| | w/o meta-initialization | | w/ meta-initialization | |
|---|---|---|---|---|
| Environment | PSNR | SSIM | PSNR | SSIM |
| Room-0 | 29.988 | 0.8184 | **30.920** | **0.8373** |
| Room-1 | 34.547 | 0.9279 | **34.871** | **0.9305** |
| Room-2 | 36.680 | 0.9560 | **37.460** | **0.9609** |
| Office-0 | 38.674 | 0.9629 | **39.290** | **0.9680** |
| Office-1 | 36.196 | 0.9427 | **36.867** | **0.9460** |
| Office-2 | 42.648 | 0.9638 | **42.665** | **0.9646** |

such as AR/VR, gaming systems, or real estate demonstrations. Depth or geometric data are less sensitive since it provides only shape outlines that are less identifiable and do not leak personal information seriously.

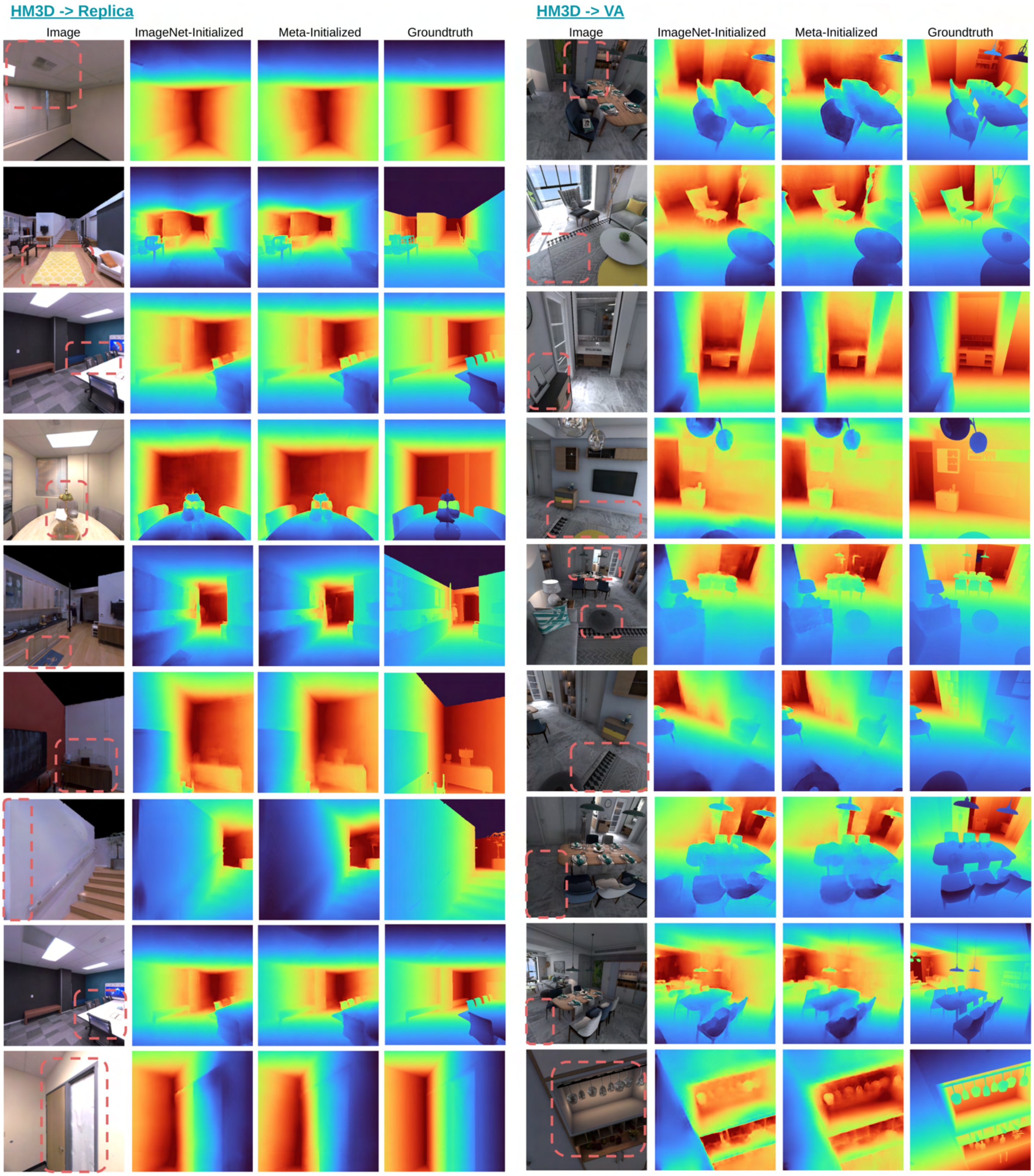
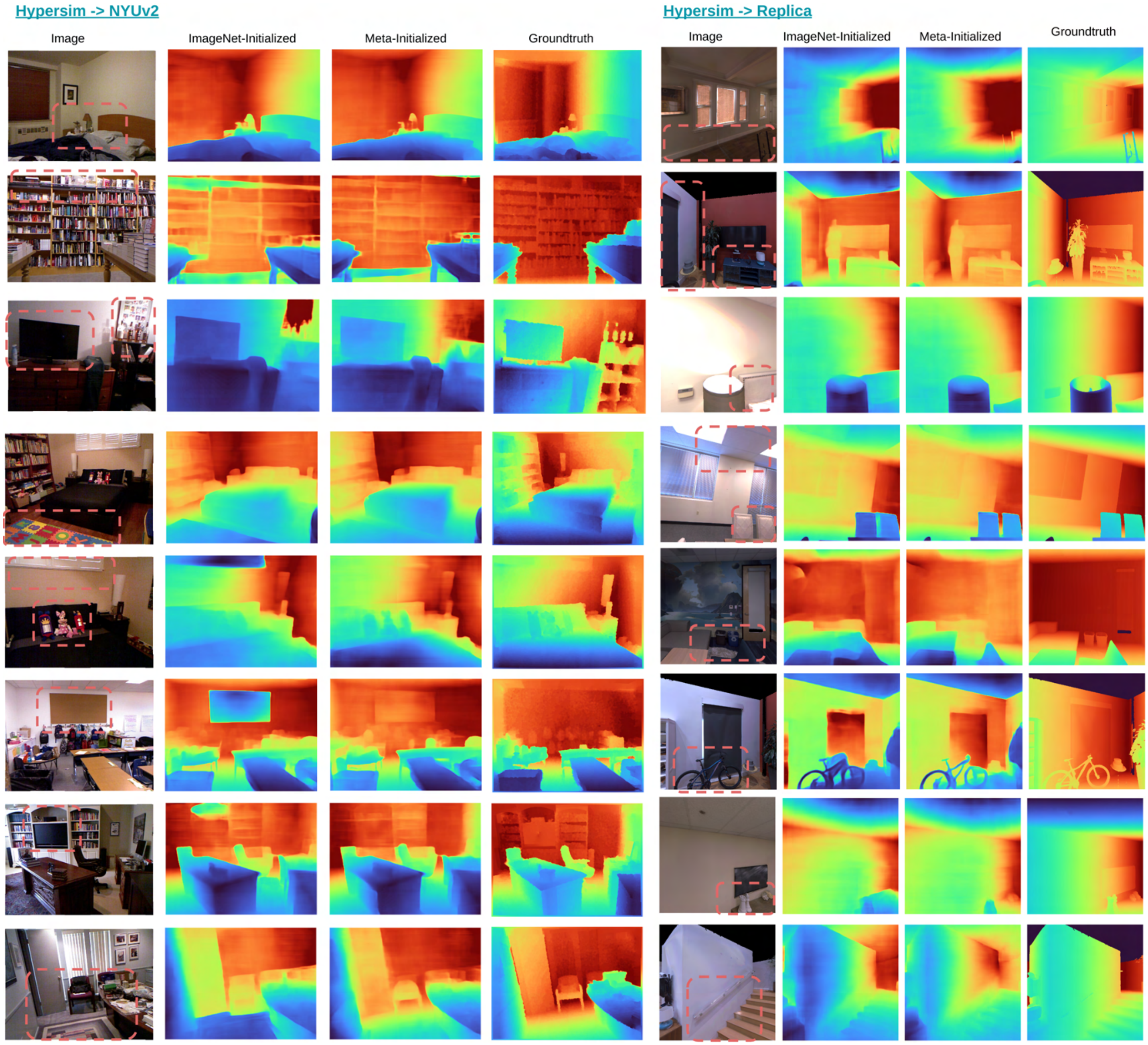


Figure S4: **Qualitative comparison on cross-dataset inference using ConvNeXt-Base.** Highlighted areas show the differences. Zoom in for the best view.


Hypersim -> NYUv2
Image
ImageNet-Initialized
Meta-Initialized
Groundtruth
Hypersim -> Replica
Image
ImageNet-Initialized
Meta-Initialized
Groundtruth

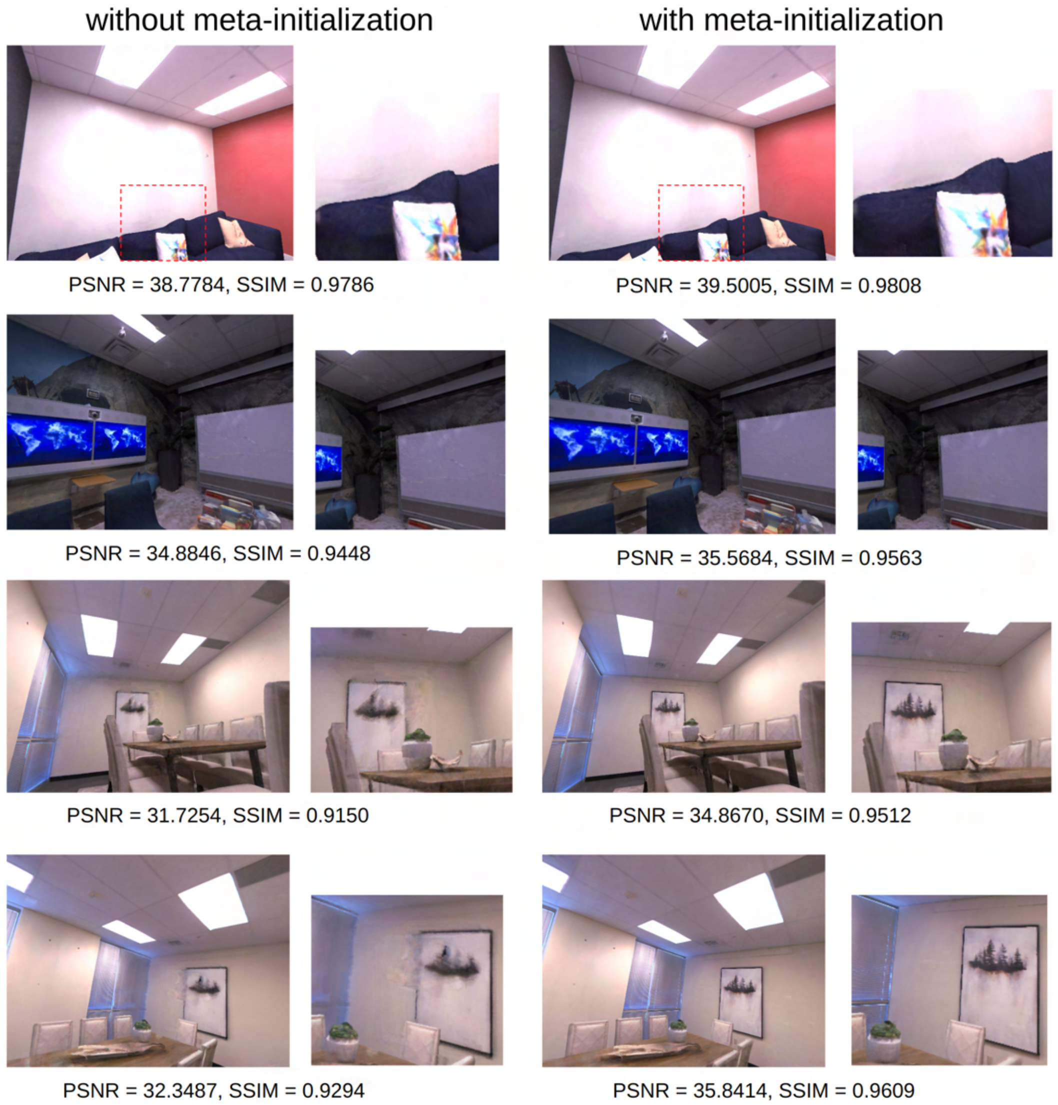

without meta-initialization
with meta-initialization
PSNR = 38.7784, SSIM = 0.9786
PSNR = 39.5005, SSIM = 0.9808
PSNR = 34.8846, SSIM = 0.9448
PSNR = 35.5684, SSIM = 0.9563
PSNR = 31.7254, SSIM = 0.9150
PSNR = 34.8670, SSIM = 0.9512
PSNR = 32.3487, SSIM = 0.9294
PSNR = 35.8414, SSIM = 0.9609

Figure S6: **Image quality comparison for NeRF rendering.** We show the quality metrics (the higher the better) under each image. Zoom in for the best view.

## References