Prism: Semi-Supervised Multi-View Stereo with Monocular Structure Priors

Alex Rich anrich@ucsb.edu
Noah Stier noahstier@ucsb.edu
Pradeep Sen psen@ucsb.edu
Tobias Höllerer holl@cs.ucsb.edu
University of California, Santa Barbara

Abstract

The promise of unsupervised multi-view stereo (MVS) is to leverage large unlabeled datasets, yet current methods underperform when training on difficult data, such as handheld smartphone videos of indoor scenes. Meanwhile, high-quality synthetic datasets are available but MVS networks trained on these datasets fail to generalize to real-world examples. To bridge this gap, we propose a semi-supervised learning framework that allows us to train on real and rendered images jointly, capturing structural priors from synthetic data while ensuring parity with the real-world domain. Central to our framework is a novel set of losses that leverages powerful existing monocular relative-depth estimators trained on the synthetic dataset, transferring the rich structure of this relative depth to the MVS predictions on unlabeled data. Inspired by perceptual image metrics, we compare the MVS and monocular predictions via a deep feature loss and a multi-scale statistical loss. Our full framework, which we call Prism, achieves large quantitative and qualitative improvements over current unsupervised and synthetic-supervised MVS networks. This is quite a useful result, opening the door to using both unlabeled smartphone videos and photorealistic synthetic datasets for training MVS networks.

1. Introduction

Multi-view stereo (MVS) is a central problem in computer vision [13, 42, 43], with applications from augmented reality to autonomous driving and robotics. While fully-supervised deep-learning-based MVS has seen great advances [4, 9, 14, 29, 31, 36, 57, 61], these methods rely heavily on accurate ground-truth 3D geometry collected using depth sensors. This is time-consuming to collect, limiting dataset size significantly compared to existing segmentation, classification, or text datasets, for example.

The promise of unsupervised multi-view stereo is to gain access to the same large amounts of training data available to other fields. However, existing unsupervised MVS training methods [5, 8, 18, 24, 27, 34, 37, 53–55, 58, 66] generally only demonstrate training on the highly-constrained, laboratory-collected DTU dataset [21]. We find that these methods cannot handle more difficult data such as handheld smartphone video of indoor scenes, often failing on reflective surfaces (Fig. 1A) or complex geometry. This significantly limits their real-world usefulness. Concurrently, high-quality synthetic datasets have emerged [1, 23, 38], but their utility for training MVS networks is unclear. Networks trained fully-supervised with these datasets do not generalize to real examples, often predicting noisy surfaces and incorrect geometry (Fig. 1B). Using basic semi-supervision (i.e., jointly training unsupervised on smartphone data and supervised on synthetic data) does help performance, but these networks also predict incorrect geometry (Fig. 1C). Most notably, none of these baseline methods learn a reasonable *structure* prior that matches the complexity of real

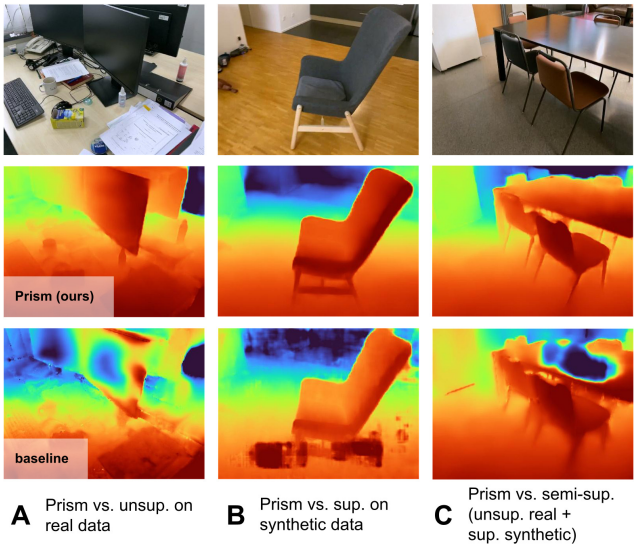

Figure 1. Using structure priors from a monocular relative-depth network, Prism effectively trains with a combination of real unlabeled smartphone video and synthetic data. It outperforms all 3 baselines: unsupervised on smartphone data (**A**), supervised on synthetic data (**B**), and semi-supervised using both (**C**). Results shown are for the ScanNet++ dataset [62]. See Sec. 4 for details.

https://github.com/alexrich021/prism

data and therefore cannot handle, for instance, textureless and reflective surfaces (A, C) or thin structures (B).

Meanwhile, recent work on diffusion-based monocular relative-depth predictors such as Marigold [22] and Lotus [15] can train robust networks on small synthetic datasets. These networks predict highly-structured but *relative* depth, i.e., depth of arbitrary scale and shift from the ground truth. Naturally, the question arises: can we transfer the structure prior these monocular networks learn on the synthetic data to MVS network predictions on real data?

To this end, we propose **Prism**, a semi-supervised learning framework that leverages both unlabeled smartphone video and synthetic data effectively to train MVS networks with high-quality structure priors (Fig. 1, middle row). Central to our framework is a novel set of losses that leverage powerful existing monocular relative-depth estimators trained on the synthetic dataset, transferring the rich structure of this relative depth to the MVS predictions on unlabeled data. Our key observation with these losses is that we can apply ideas from RGB perceptual metrics to learn structure from relative depth. Specifically, we apply two losses. First, inspired by LPIPS [67], we extract deep features from the monocular and MVS predictions via a pre-trained RGB feature extractor and enforce them to be close. Second, inspired by multi-scale structural similarity (SSIM) [47], we encourage the statistics of the monocular and MVS predictions to be similar at multiple resolutions. Interestingly, these concepts from perceptual image metrics apply readily to depth maps, allowing us to transfer structural priors from relative-depth networks and far outperforming pixel-wise ℓ_1 and single-scale SSIM monocular losses.

In addition to these monocular losses, Prism uses unsupervised losses on the smartphone data and supervised losses on synthetic examples. We demonstrate joint training on ScanNet++ iPhone videos [62] and the Hypersim synthetic dataset [38]. Our full framework achieves large quantitative and qualitative improvements in depth prediction over all baselines on the ScanNet++ test set [62]. These results generalize to ARKitScenes [2], with Prism again exceeding all baselines both quantitatively and qualitatively on all metrics. We also note Prism is completely agnostic to the MVS architecture. Our contributions are as follows:

1. We propose a deep feature loss and a multi-scale statistical loss for learning structure priors from relative depth and demonstrate their superior performance compared with existing monocular losses.
2. We design Prism, a semi-supervised learning framework which uses our monocular structure priors to train MVS networks on unlabeled real and labeled synthetic data.
3. We demonstrate that Prism outperforms all baseline methods: unsupervised on real data, supervised on synthetic data, and semi-supervised on both.

Prism opens the door to using both real-world videos and photorealistic synthetic datasets jointly for training MVS networks, taking a positive step towards more extensive training data for 3D reconstruction.

2. Related Work

Monocular prior losses: Exploiting monocular networks as a training signal is common in many fields. In sparse-view novel-view synthesis and neural-implicit 3D reconstruction, monocular cues are applied as pixel-wise constraints to regularize ambiguous geometry, using ℓ_1 or ℓ_2 losses [7, 17, 44, 64] sometimes modulated by uncertainty estimations [39, 52], multi-view checks [45], or both [6]. In monocular depth training, a teacher-student paradigm is common. Knowledge from the teacher network is transferred via pseudo-labels [50, 59, 60]. Pixel-wise supervision is most common here too [59, 60], though DistDepth [50] proposes an SSIM loss, taking a step away from pixel-wise supervision. In contrast to all of these methods, we transfer larger, patch-level *structure* from monocular depth using both multi-scale statistics and deep features. We show this is more effective than pixel-wise or single-scale SSIM approaches, making our work highly relevant. Furthermore, to the best of our knowledge, monocular prior losses have not been applied to MVS depth prediction.

Fully-supervised MVS: Methods which train on real data are constrained by a reliance on ground-truth 3D geometry [4, 9, 11, 16, 19, 29, 30, 36, 41, 51, 61, 63]. Recently, MVSAnywhere [20] and the two-view Foundation-Stereo [49] show that significantly scaling synthetic data reduces the domain gap. Here, our work is complimentary. First, we demonstrate only a small synthetic set is sufficient. Second, we allow these methods to train on real data with no architectural changes, further closing the domain gap.

Unsupervised MVS: The field has largely settled on a combination of photometric, depth-smoothness, and augmentation-consistency losses [5, 8, 18, 24, 27, 37, 53, 54], sometimes with pseudo-labeling [10, 34, 55, 58]. We show that these methods do not work on smartphone video, and propose a semi-supervised method that does. Our framework allows these techniques to be widely applicable.

Semi-supervised MVS: There is extremely limited work here. To the best of our knowledge, it all explores the single-dataset setting, either assuming sparse ground-truth points are provided [25, 65] or only a subset of images have ground-truth depth [56]. We are the first work exploring the important area of multi-dataset semi-supervised MVS.

Monocular depth: While many works focus on scaling the training set [3, 12, 28, 35, 59, 60], Marigold [22] and Lotus [15] have proposed fine-tuning Stable Diffusion v2 [40] for depth prediction on a small synthetic dataset. We use these methods to capture structural priors from minimal synthetic data. We show that we can transfer this prior to MVS predictions via carefully-constructed losses.

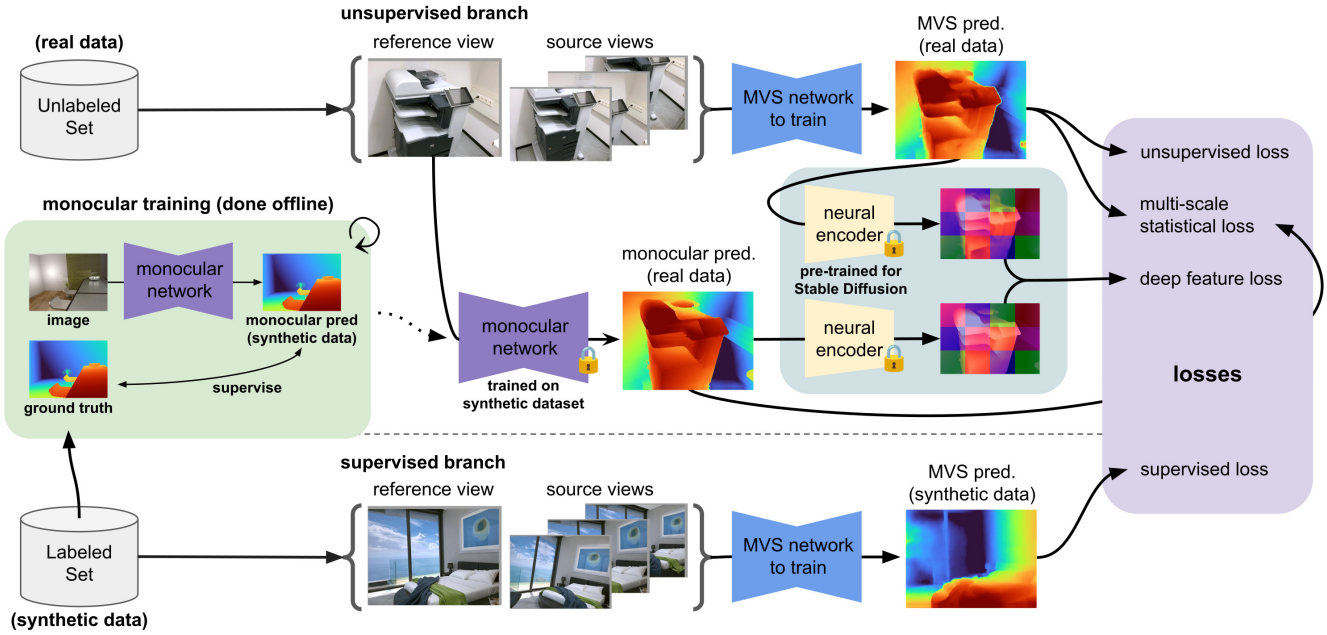

Figure 2. **Overview of Prism framework for semi-supervised MVS.** We leverage both real unlabeled smartphone data and labeled synthetic data to train MVS networks. Our central idea is to (1) train an existing monocular relative-depth prediction network on the synthetic set in order to capture high-quality structure priors and (2) teach the MVS network to use these structural priors on the unlabeled set via losses inspired by perceptual image metrics. In addition to these monocular losses, we also utilize unsupervised losses on the unlabeled real examples and supervised losses on the synthetic examples.

3. Method

In this section we describe our novel Prism framework for semi-supervised training of MVS networks (Fig. 2). Our method takes as input an unlabeled and labeled set of images. Each is assumed to have known camera parameters, and the labeled set is assumed to have ground-truth depth information. In our experiments, we use smartphone images and synthetic data for these sets. Prior to training the MVS network, we train a monocular depth prediction network on the labeled set. At each training iteration, reference images $\mathbf{I}^U$ and $\mathbf{I}^L$ are sampled from the unlabeled and labeled sets respectively, along with a set of source images and camera parameters and, for the labeled set, the ground-truth depth $\mathbf{D}^L_{gt}$. The MVS network to be trained makes predictions $\mathbf{D}^U$ and $\mathbf{D}^L$ on the unlabeled and labeled samples respectively. The loss is then computed as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{mono}} + \lambda_2 \mathcal{L}_{\text{unsup}} + \lambda_3 \mathcal{L}_{\text{sup}}, \qquad (1)$$

where $\mathcal{L}_{\text{mono}}$ uses the monocular network to compute a loss on $\mathbf{D}^U$, $\mathcal{L}_{\text{unsup}}$ uses the standard photometric, depth-smoothness, and augmentation-consistency supervision from DIV loss [37] to compute a loss on $\mathbf{D}^U$, and $\mathcal{L}_{\text{sup}}$ computes a supervised loss comparing $\mathbf{D}^L$ and $\mathbf{D}^L_{gt}$. λ_1, λ_2, and λ_3 weight the contributions of each loss. Other than differentiability, we make no assumptions on the MVS network to be trained.

3.1. Monocular Structure Priors

As noted in Sec. 1 and Fig. 1, neither $\mathcal{L}_{\text{unsup}}$ nor $\mathcal{L}_{\text{sup}}$ help the MVS network learn reasonable structural priors for real examples. To capture these priors from the labeled synthetic data, we train a monocular depth predictor. We then teach the MVS network to use these structural priors on the unlabeled set via losses inspired by perceptual image metrics, specifically a deep feature loss inspired by LPIPS [67] and a statistical loss inspired by multi-scale SSIM [47].

Monocular network, prediction, and normalization: For our monocular network, we take advantage of the methods which fine-tune diffusion models for relative-depth prediction on a small synthetic dataset [15, 22]. Specifically, we train Marigold [22] on the labeled synthetic set prior to MVS training. Then, given the reference image $\mathbf{I}^U$ on the unlabeled set, we make a relative-depth prediction $\mathbf{D}^U_*$ using this monocular network. For later use with the deep feature loss, we normalize $\mathbf{D}^U_*$ to be in the range $[0, 1]$:

$$\bar{\mathbf{D}}^U_* = \frac{\mathbf{D}^U_* - q_2}{q_{98} - q_2}, \qquad (2)$$

where q_a is the a^{th} quantile of $\mathbf{D}^U_*$. $\bar{\mathbf{D}}^U_*$ is affine-invariant, i.e., it is predicted up to a scale and shift ambiguity s, t from the ground truth. Following Ranftl *et al.* [35], we compute s, t that align $\bar{\mathbf{D}}^U_*$ with the MVS prediction $\mathbf{D}^U$ as

$$(s, t) = \arg\min_{s,t} \sum_{\mathbf{p}} (s\bar{\mathbf{D}}^U_*(\mathbf{p}) + t - \mathbf{D}^U(\mathbf{p}))^2, \qquad (3)$$

which has an analytic solution.

Deep feature loss: We want to transfer patch-level structure from the monocular depth prediction. Inspired by LPIPS [67], we find deep features from pre-trained feature extractors are an extremely effective method of doing this. First, these extractors naturally have a large receptive field. Second, they are influenced more by higher-level structures than by pixel-level variation.

Specifically, we compare deep embeddings of $\mathbf{D}^U$ and $\bar{\mathbf{D}}^U_*$ using a pre-trained feature extractor. This feature extractor expects 3-channel images with values in the range $[0, 1]$, so we first align $\mathbf{D}^U$ with $\bar{\mathbf{D}}^U_*$ as $\bar{\mathbf{D}}^U = \frac{1}{s}(\mathbf{D}^U - t)$, putting it approximately in the correct range. We then duplicate both $H \times W$ depth maps $\bar{\mathbf{D}}^U$ and $\bar{\mathbf{D}}^U_*$ 3 times to form a $H \times W \times 3$ depth "image." Using the pre-trained feature extractor, we compute $H' \times W' \times C$ deep embeddings $\mathbf{F}$ and $\mathbf{F}_*$ of these depth images. Finally, we normalize in the channel dimension, denoting the normalized embeddings as $\bar{\mathbf{F}}$ and $\bar{\mathbf{F}}_*$, and take the mean ℓ_2 distance between normalized embeddings as our feature loss ℓ_{feat}:

$$\ell_{\text{feat}} = \frac{1}{H'W'} \sum_{\mathbf{p}} \|\bar{\mathbf{F}}(\mathbf{p}) - \bar{\mathbf{F}}_*(\mathbf{p})\|. \tag{4}$$

While we could compare multiple feature scales like LPIPS we find that for depth maps, unlike RGB images, comparing only the deepest feature embedding gives the best results. We tested a variety of pre-trained feature extractors, and find that the encoder from Stable Diffusion v2 [40] gives the largest performance boost. During initial development, we also tested passing our depth images directly to LPIPS, and found it to cause artifacts.

Statistical loss: In addition to comparing deep features, we also find that comparing patch-wise statistics of $\mathbf{D}^U$ and $\mathbf{D}^U_*$ helps transfer the monocular structure prior. There are two obvious options: SSIM [46, 48] and MS-SSIM [47]. In both cases, the loss can be taken as the negative similarity of aligned depth maps:

$$\ell_{\text{ssim}} = 1 - f_{\text{sim}}(\mathbf{D}^U, s\bar{\mathbf{D}}^U_* + t), \tag{5}$$

where f_{sim} denotes SSIM or MS-SSIM. Single-scale SSIM does improve results slightly, but is limited to a single receptive field size. MS-SSIM is computed as the product of patch-wise statistics for several patch scales. This has the added benefit of multiple receptive fields increasing in size; however, we find the multiplication operation for combining the scale-wise statistics can cause instability in the loss term. If one term is small, that term dominates and the other terms are ignored. We find a summation operation to be more stable. We therefore define a summation-based MS-SSIM as the normalized sum of single-scale SSIM:

$$f_{\text{sim}}(\mathbf{x}, \mathbf{y}) = \frac{1}{L} \sum_{l=1}^{L} \text{SSIM}(\downarrow_l (\mathbf{x}), \downarrow_l (\mathbf{y})). \tag{6}$$

where $\downarrow_l (*)$ applies a low-pass filter and then downsamples to scale l. Our loss ℓ_{ssim} is then computed as in Eq. 5, the negative similarity. Eq. 6 simply computes SSIM on the image pyramid so, to distinguish it from standard MS-SSIM, we refer to it as "pyramid SSIM" (P-SSIM).

Full monocular loss term: Our final monocular loss is

$$\mathcal{L}_{\text{mono}} = \ell_{\text{feat}} + \alpha \ell_{\text{ssim}}, \tag{7}$$

where ℓ_{feat} and ℓ_{ssim} are given in Eqs. 4 and 5 and α weights the terms. We set $\alpha = 1.0$ and use $L = 4$ levels.

3.2. Additional Losses

In addition to $\mathcal{L}_{\text{mono}}$, we also compute an unsupervised loss $\mathcal{L}_{\text{unsup}}$ on $\mathbf{D}^U$ and supervised loss $\mathcal{L}_{\text{sup}}$ on $\mathbf{D}^L$.

Unsupervised loss: We use the standard photometric, depth-smoothness, and augmentation-consistency losses from the literature, opting for the exact DIV loss formulation recently proposed [37]. The unsupervised loss is

$$\mathcal{L}_{\text{unsup}} = \alpha_1 \mathcal{L}_{\text{photo}} + \alpha_2 \mathcal{L}_{\text{ssim}} + \alpha_3 \mathcal{L}_{\text{sm}} + \alpha_4 \mathcal{L}_{\text{aug}}. \tag{8}$$

We omit the details for space, and refer the reader to Rich *et al*. [37]. We use the same hyperparameters, multiplying α_3 and α_4 by a factor of 100 to account for dataset scale differences between DTU and ScanNet++. Our final hyperparameters are $\alpha_1 = 12.0$, $\alpha_2 = 6.0$, $\alpha_3 = 18.0$, and $\alpha_4 = 1.0$.

Supervised loss: Inspired by Sayed *et al*. [41], we use a regression, multi-scale gradient, and normal loss:

$$\mathcal{L}_{\text{sup}} = \ell_{\text{regr}} + \ell_{\text{grad}} + \ell_{\text{normals}}. \tag{9}$$

The regression loss ℓ_{regr} computes log ℓ_1 error:

$$\ell_{\text{regr}} = \frac{1}{HW} \sum_{\mathbf{p}} |\log \mathbf{D}^L(\mathbf{p}) - \log \mathbf{D}^L_{gt}(\mathbf{p})|. \tag{10}$$

The multi-scale gradient loss ℓ_{grad} is

$$\ell_{\text{grad}} = \sum_{l=1}^{4} \sum_{\mathbf{p}} |\nabla \downarrow_l (\mathbf{D}^L(\mathbf{p})) - \nabla \downarrow_l (\mathbf{D}^L_{gt}(\mathbf{p}))|, \tag{11}$$

where ∇ is the first-order spatial gradient, and $\downarrow_l (*)$ applies a low-pass filter followed by downsampling to scale l, as before. For ℓ_{normals} we first extract normals $\mathbf{N}$ and $\mathbf{N}_{gt}$ from $\mathbf{D}^L$ and $\mathbf{D}^L_{gt}$ using the known camera intrinsics, then compute the loss as

$$\ell_{\text{normals}} = \frac{1}{2HW} \sum_{\mathbf{p}} 1 - \mathbf{N}(\mathbf{p})^T \mathbf{N}_{gt}(\mathbf{p}), \tag{12}$$

where the 2 in the denominator normalizes the loss magnitude to $[0, 1]$. In our experiments, we use CasMVSNet-style cost-volume-based networks [14]. We find that specifically for the indoor setting we train and test on, these losses are superior to the ℓ_1 regression or probability-based classification losses commonly used for these networks. For the lower-resolution depth maps produced by our network, we upsample to full resolution and supervise only with ℓ_{regr}.

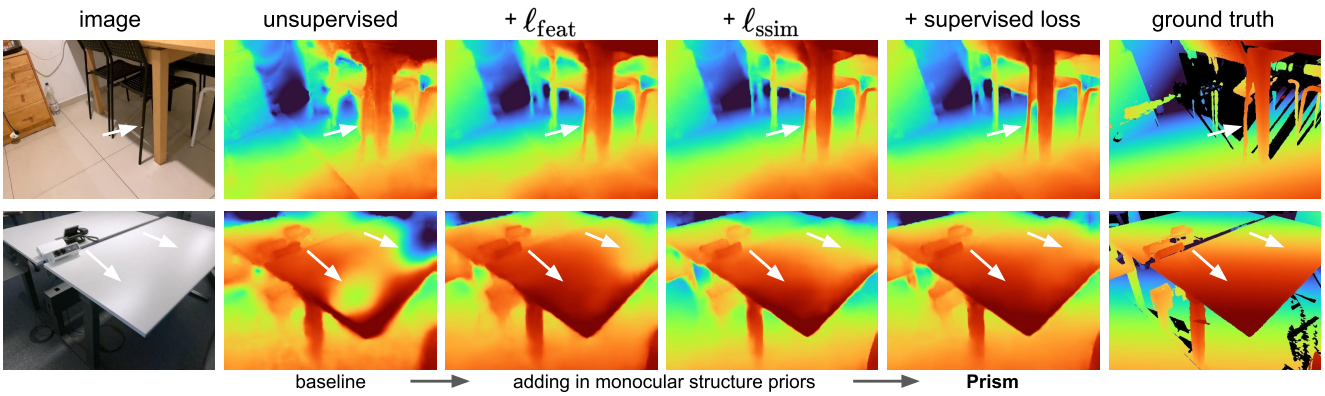

Figure 3. **Visual Ablation Study.** Each component of Prism interacts constructively, helping bring out fine detail (top row) and global structure (bottom row). The monocular structure prior significantly helps in the case of textureless/reflective surfaces. The supervised loss helps add a final smoothness to flat surfaces and sharpness to object boundaries. See Sec. 4.5 for details and Table 2 for quantitative results.

3.3. Final Details

We set the monocular, unsupervised, and supervised weights as $\lambda_1 = 10, \lambda_2 = 1$, and $\lambda_3 = 10$, chosen to balance the relative magnitude of the losses. It is likely better hyperparameters can be found using a validation set; however, we find these to be quite effective. Finally, we only activate the monocular loss after the first epoch of training, allowing for initial convergence so the scale and shift estimation (Eq. 3) is reasonable.

4. Experiments

4.1. Implementation Details

In our main comparisons, we use the CasMVSNet-style network from DIV-MVS [37] as our MVS network. In our ablation study, we include results with MVSFormer-P [4] as our MVS network. We implement Prism in PyTorch [33] and make heavy use of Open3D [68] for visualization.

Training data: For our unlabeled smartphone dataset, we use the ScanNet++ iPhone video training split [62] which consists of handheld capture of 230 indoor scenes. We sample every 10 frames from the iPhone videos. For our labeled synthetic dataset, we use the Hypersim training split [38], which consists of high-quality, photo-realistic renderings of 365 artist-created indoor spaces with corresponding depth ground truth. We resize all images to 384×512.

Baselines: We compare against 3 main baselines. The first is our base unsupervised method, DIV-MVS [37]. The second is our base supervised method, i.e., our CasMVSNet-style network trained using $\mathcal{L}_{\text{sup}}$ on synthetic data. The third is a base semi-supervised method trained both unsupervised on real data and supervised on synthetic data but without monocular structure priors (i.e., using only $\mathcal{L}_{\text{unsup}}$ and $\mathcal{L}_{\text{sup}}$ but not $\mathcal{L}_{\text{mono}}$). Note that for all of these baselines, we keep the MVS network constant and change *only* the training. Furthermore, to the best of our knowledge, our semi-supervised baseline represents a novel combination of joint supervised and unsupervised MVS training. Due to its good performance, we hope that it may be useful for future work. In addition to these main baselines, we also compare against two other unsupervised methods from the literature, RC-MVSNet [5] and CL-MVSNet [53]. Note for DIV, RC, and CL we re-train on ScanNet++ for fair comparison as we found the DTU weights generalize poorly to ScanNet++.

Training parameters: All pipelines are trained from scratch for 160k steps. We use the Adam optimizer [26], with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The learning rate is halved after 100k, 120k, and 140k steps. We use a batch size of 4, which requires 4 NVIDIA RTX 3090 GPUs. For all unsupervised baselines, we use the hyperparameters detailed in Sec. 3.2. Following existing work [5, 37, 54], we double the augmentation-consistency weight every 20k steps from step 10k until 90k.

4.2. Evaluation Details

Testing data: For our main comparisons, we use the ScanNet++ iPhone semantic test set [62], which consists of 50 complex and challenging indoor scenes. To test the generalization ability of Prism, we additionally test on the ARKitScenes 3D object detection validation set [2] with *no finetuning on additional data*. This dataset consists of iPhone images with corresponding depth maps from ARKit. We randomly select 50 scenes from the subset of the validation set where "up" in the image is aligned with gravity.

Protocols: We use 5 images for depth prediction. For both datasets, we resize the images to 384×512 and make a depth prediction at the same resolution. We set the minimum and maximum depth planes to 25cm and 5m respectively for both datasets. For quantitative comparison, we use the standard metrics from Murez *et al*. [32].

4.3. Results

Overall, we see Prism outperforms all baselines on all datasets tested (Sec. 4.4), each component of Prism inter-

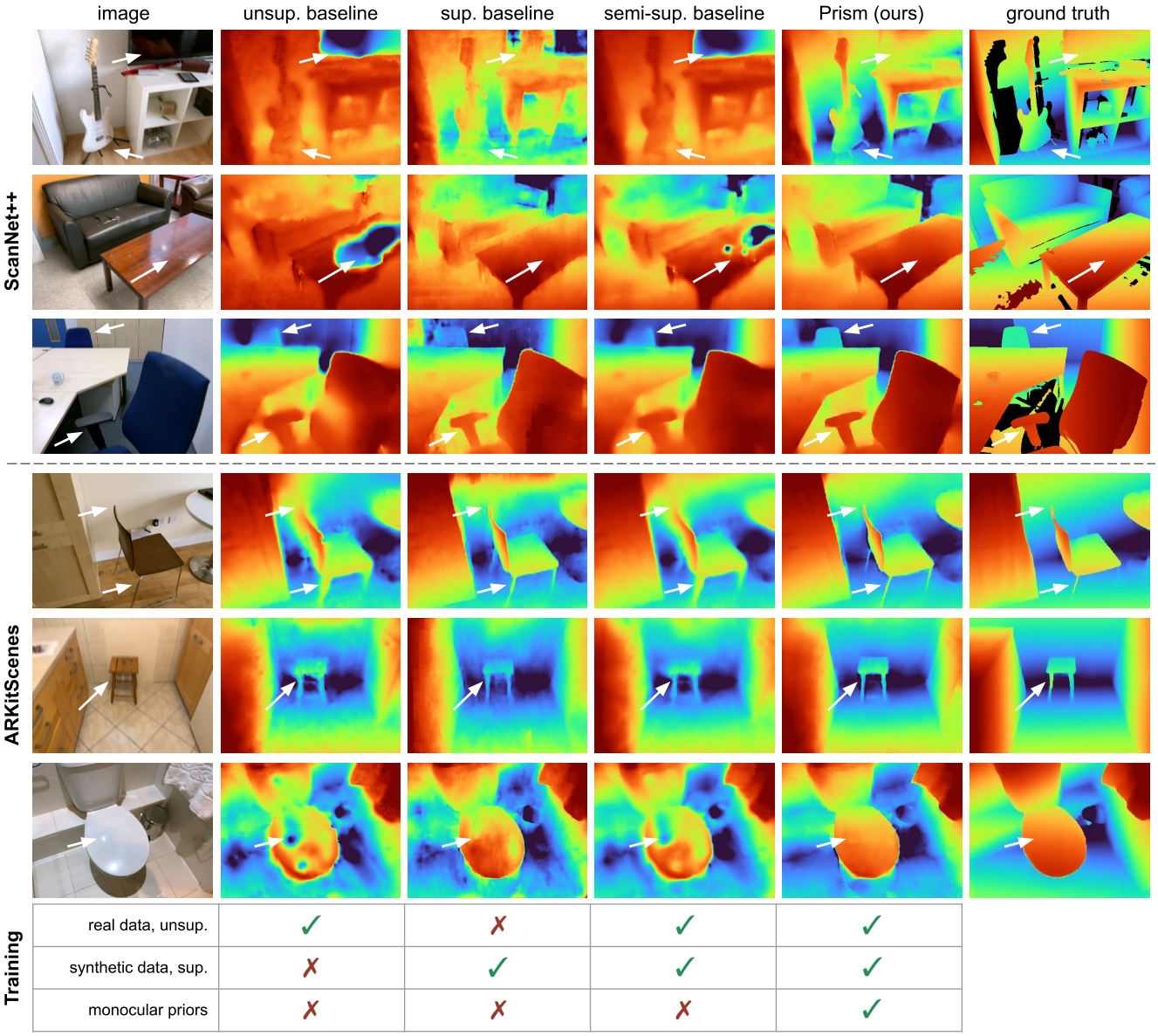

Figure 4. **Qualitative Results.** Prism outperforms all baselines, producing sharp and accurate depth maps with excellent global structure and fine-grained local detail. With the arrows, we indicate hard cases where Prism performs well: textureless and reflective surfaces (rows 1, 2, 6) and thin structures (rows 1, 3, 4, 5). In our ablation study, we find the monocular losses largely help with textureless and reflective surfaces, while all components interact constructively to improve thin structures.

acts constructively to improve results (Sec. 4.5), and Prism works seamlessly with other MVS networks (Sec. 4.5).

4.4. Main Comparisons

See Table 1 and Fig. 4 for quantitative and qualitative results on ScanNet++ and ARKitScenes. Prism outperforms all competing baselines on all metrics on both datasets (Table 1). In particular, we improve metrics by over **10%** in nearly all cases compared to our semi-supervised baseline, confirming the effectiveness of our monocular structure priors. We highlight that this is true not only for ScanNet++, whose training data we use for our unlabeled set, but also for ARKitScenes. This indicates Prism learns a general structure prior that transfers across datasets. We also note that when comparing exclusively to published work, i.e., the unsupervised and synthetic-supervised baselines, we improve many metrics by over **20** to **30%**.

Qualitatively, Prism produces depth maps with coherent global structure and sharp local detail (Fig. 4). In particular, we find Prism noticeably improves performance in two hard cases. First, Prism predicts accurate depth for textureless and reflective surfaces while most competing baselines fail.

	RC [5]	CL [53]	DIV [37]	Cas [14]	DIV [37] + sup.	**Prism** (ours)	% diff. vs. baselines vs. unsup.	vs. sup.	vs. semi-sup.
TRAINING			(unsup. base)	(sup. base)	(semi-sup. base)				
real data, unsup.	✓	✓	✓	✗	✓	✓			
synthetic, sup.	✗	✗	✗	✓	✓	✓			
monocular priors	✗	✗	✗	✗	✗	✓			
SCANNET++									
Abs-rel ↓	0.124	0.126	0.123	0.118	0.100	**0.090**	-26.8%	-23.7%	-10.0%
Abs-diff ↓	0.207	0.222	0.200	0.215	0.179	**0.158**	-21.0%	-26.5%	-11.7%
Abs-inv ↓	0.090	0.091	0.088	0.133	0.077	**0.068**	-22.7%	-48.9%	-11.7%
Sq-rel ↓	0.082	0.082	0.086	0.086	0.066	**0.052**	-39.5%	-39.5%	-21.2%
$\delta < 1.25$ ↑	0.840	0.829	0.848	0.831	0.873	**0.895**	5.5%	7.7%	2.5%
ARKITSCENES									
Abs-rel ↓	0.193	0.182	0.182	0.129	0.127	**0.115**	-36.8%	-10.9%	-9.4%
Abs-diff ↓	0.215	0.212	0.203	0.183	0.165	**0.148**	-27.1%	-19.1%	-10.3%
Abs-inv ↓	0.144	0.137	0.138	0.169	0.108	**0.100**	-27.5%	-40.8%	-7.4%
Sq-rel ↓	0.154	0.139	0.163	0.077	0.096	**0.069**	-57.7%	-10.4%	-28.1%
$\delta < 1.25$ ↑	0.812	0.815	0.824	0.829	0.873	**0.890**	8.0%	7.4%	1.9%

Table 1. **Main Comparisons.** Depth prediction results on ScanNet++ [62] and ARKitScenes [2]. We compare with several unsupervised methods (RC, CL, DIV), as well as a fully-supervised baseline trained on the synthetic Hypersim dataset [38], and a semi-supervised baseline without our novel monocular structure priors. Prism outperforms all baselines on all metrics on both datasets, demonstrating the efficacy of our monocular structure priors. In particular, we find metrics are improved by over **10%** in nearly all cases vs. our semi-supervised baseline. We outperform the published unsupervised and synthetic-supervised baselines by over **20** to **30%** on many metrics. Note that for RC, CL, and DIV we re-train on ScanNet++ for fair comparison.

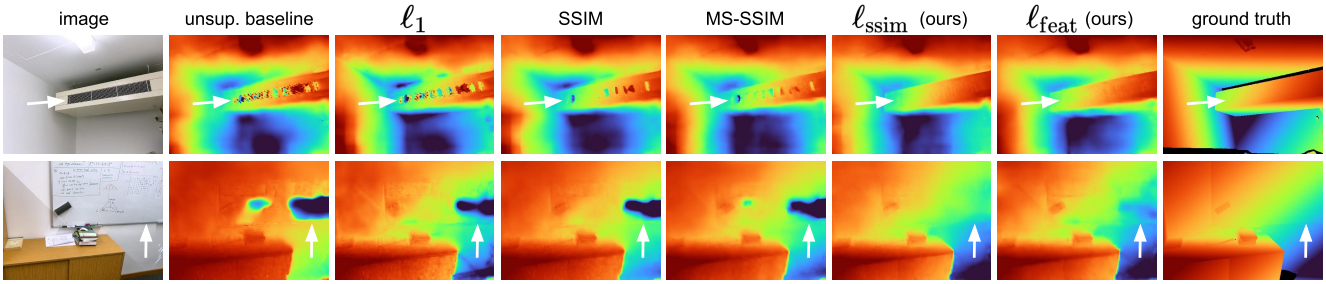

Figure 5. **Visual Ablation Study.** We isolate the effect of various monocular losses. Only our multi-scale loss ℓ_{ssim} and our deep feature loss ℓ_{feat} can handle confusing geometry (top row) and textureless surfaces (bottom row). See Table 2 for quantitative results.

Second, Prism predicts thin structures with high precision while competing baselines tend to predict blurry, overly-smoothed object boundaries. These results further highlight the effectiveness of our monocular structure priors.

4.5. Ablation Study

In Table 2, we analyze each component of Prism, showing our monocular structure priors are critical to our performance boost and each component interacts constructively. Note that rows 0, 1, and 2 of the table correspond to our unsupervised, supervised, and semi-supervised base methods. In Table 3, we demonstrate that Prism works seamlessly with other MVS networks, indicating general applicability of Prism not tied to a specific network architecture.

Basic semi-supervision helps: In Table 2, we show that basic semi-supervision (row 2) does provide a nice quantitative boost. However, as detailed in Sec. 4.4 and shown in Figs. 1 and 4, it fails to learn a structure prior, often failing on textureless/reflective surfaces and thin structures.

Our monocular structure priors are critical: In rows 3 through 7 of Table 2, we show our monocular losses outperform existing methods. The baseline method for these rows is unsupervised (row 0). We test 3 alternative options: the commonly used pixel-wise ℓ_1 loss (row 3), as well as SSIM (row 4) and MS-SSIM (row 5). ℓ_1 and SSIM provide limited benefit. Only with MS-SSIM does performance begin to noticeably improve, indicating that reasonably-sized receptive fields help these losses. Our ℓ_{ssim} (row 6, Eq. 5) improves on the standard MS-SSIM, boosting performance in most cases while maintaining performance otherwise. Our deep-feature loss (row 7, ℓ_{feat}, Eq. 4) provides a similar boost as ℓ_{ssim} and outperforms MS-SSIM on every metric.

In Fig. 5, we isolate the effect of each monocular loss. Only our proposed losses perform well on confusing ge-

	losses			ScanNet++				ARKitScenes			
	unsup.	sup.	mono.	Abs-rel ↓	Abs-diff ↓	Abs-inv ↓	$\delta < 1.25$ ↑	Abs-rel ↓	Abs-diff ↓	Abs-inv ↓	$\delta < 1.25$ ↑
0	✓			0.123	0.200	0.088	0.848	0.182	0.203	0.138	0.824
1		✓		0.118	0.215	0.133	0.831	0.129	0.183	0.169	0.829
2	✓	✓		0.100	0.179	0.077	0.873	0.127	0.165	0.108	0.873
3	✓		L1	0.110	0.206	0.083	0.852	0.150	0.188	0.126	0.841
4	✓		SSIM	0.107	0.193	0.081	0.859	0.148	0.181	0.123	0.850
5	✓		MS-SSIM	0.100	0.181	0.075	0.875	0.136	0.171	0.114	0.867
6	✓		ℓ_{ssim}	0.096	0.178	0.074	0.880	0.127	0.165	0.114	0.867
7	✓		ℓ_{feat}	0.099	0.171	0.072	0.884	0.135	0.162	0.110	0.875
8	✓		$\ell_{\text{feat}} + \ell_{\text{ssim}}$	0.093	0.165	0.070	0.889	0.123	0.154	0.108	0.879
9	✓	✓	$\ell_{\text{feat}} + \ell_{\text{ssim}}$	**0.090**	**0.158**	**0.068**	**0.895**	**0.115**	**0.148**	**0.100**	**0.890**

Table 2. **Loss Ablation Study.** We show each component of Prism interacts constructively and our monocular structure priors noticeably outperform existing options. See Sec. 4.5 for details and Figs. 3 and 5 for qualitative results.

	unsup. base + MVSFormer	sup. base + MVSFormer	semi-sup. base + MVSFormer	**Prism** (ours) + MVSFormer	% diff. vs. baselines vs. unsup.	vs. sup.	vs. semi-sup.
TRAINING							
real data, unsup.	✓	✗	✓	✓			
synthetic, sup.	✗	✓	✓	✓			
monocular priors	✗	✗	✗	✓			
SCANNET++							
Abs-rel ↓	0.113	0.110	0.099	**0.083**	-26.5%	-24.5%	-16.2%
Abs-diff ↓	0.201	0.202	0.177	**0.151**	-24.9%	-25.2%	-14.7%
Abs-inv ↓	0.084	0.123	0.074	**0.063**	-25.0%	-48.8%	-14.9%
Sq-rel ↓	0.074	0.076	0.062	**0.042**	-43.2%	-44.7%	-32.3%
$\delta < 1.25$ ↑	0.849	0.846	0.876	**0.905**	6.6%	7.0%	3.3%

Table 3. **Network Ablation Study.** We use the popular MVSFormer-P [4] as the MVS network for Prism and all baselines instead of CasMVSNet. We find Prism works seamlessly with the MVSFormer model, noticeably improving results against all competing baselines. This indicates general applicability of Prism not tied to a specific network architecture. See Sec. 4.5 for details.

ometry (top row) or textureless/reflective surfaces (bottom row), further confirming our choices.

Prism is greater than the sum of its parts: In rows 8 and 9 of Table 2 and Fig. 3, we demonstrate that each component of Prism interacts constructively. In row 8, we show using ℓ_{ssim} and ℓ_{feat} together boosts performance on every metric. In row 9, we show the supervised loss improves performance beyond both semi-sup. (row 2) and unsup. with monocular priors (row 8). In Fig. 3, we demonstrate that each component cumulatively improves performance on thin structures and textureless/reflective surfaces.

Prism works seamlessly with other MVS networks: In our main experiments, we used the CasMVSNet-style base network from DIV-MVS for all baselines. We chose this model for fair comparison with existing unsupervised MVS methods; however, more advanced MVS networks have since been proposed. In Table 3, we include results instead using the popular and more advanced MVSFormer-P [4] network for Prism and all baselines.

We find that Prism works seamlessly with MVSFormer. The trends mirror the CasMVSNet results exactly, with Prism improving metrics by over **10%** in nearly all cases vs. our semi-supervised baseline with MVSFormer and **20** to **30%** in nearly all cases vs. our unsupervised and synthetic-supervised baselines with MVSFormer. This is an important result, indicating general applicability of Prism not tied to a specific MVS network architecture.

5. Conclusions

We have proposed Prism, a novel semi-supervised learning framework that allows us to train MVS networks on real and rendered images jointly. Central to our framework is a novel set of losses that leverages powerful existing monocular relative-depth estimators trained on the synthetic dataset, transferring the rich *structure* of this relative depth to the MVS predictions on unlabeled data. Inspired by perceptual image metrics, these losses consist of a deep feature loss and a multi-scale statistical loss. We have demonstrated that both outperform existing monocular losses while also interacting constructively. Our work bridges the gap between training on real-world RGB videos and photorealistic synthetic datasets, taking a positive step towards more extensive training data for 3D reconstruction.

References

[1] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision*, 2024. 1

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *Advances in Neural Information Processing Systems*, 2021. 2, 5, 7

[3] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth Pro: Sharp monocular metric depth in less than a second. *arXiv*, 2024. 2

[4] Chenjie Cao, Xinlin Ren, and Yanwei Fu. MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth. *Transactions on Machine Learning Research*, 2022. 1, 2, 5, 8

[5] Di Chang, Aljaž Božič, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. RC-MVSNet: Unsupervised multi-view stereo with neural rendering. In *European Conference on Computer Vision*, 2022. 1, 2, 5, 7

[6] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. VCR-GauS: View consistent depth-normal regularizer for gaussian surface reconstruction. *Advances in Neural Information Processing Systems*, 2024. 2

[7] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3D gaussian splatting in few-shot images. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2024. 2

[8] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. MVS2: Deep unsupervised multi-view stereo with multi-view symmetry. In *International Conference on 3D Vision*, 2019. 1, 2

[9] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. TransMVSNet: Global context-aware multi-view stereo network with transformers. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

[10] Yikang Ding, Qingtian Zhu, Xiangyue Liu, Wentao Yuan, Haotian Zhang, and Chi Zhang. KD-MVS: Knowledge distillation based self-supervised learning for multi-view stereo. In *European Conference on Computer Vision*, 2022. 2

[11] Arda Düzçeker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2

[12] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *International Conference on Computer Vision*, pages 10786–10796, 2021. 2

[13] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision*, 2015. 1

[14] Xiaodong Gu, Zhiwen Fan, Zuozhuo Dai, Siyu Zhu, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 4, 7

[15] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv*, 2024. 2, 3

[16] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *International Conference on Computer Vision*, 2019. 2

[17] Shoukang Hu, Kaichen Zhou, Kaiyu Li, Longhui Yu, Lanqing Hong, Tianyang Hu, Zhenguo Li, Gim Hee Lee, and Ziwei Liu. ConsistentNeRF: Enhancing neural radiance fields with 3D consistency for sparse view synthesis. In *arXiv*, 2023. 2

[18] Baichuan Huang, Hongwei Yi, Can Huang, Yijia He, Jingbin Liu, and Xiao Liu. M3VSNet: Unsupervised multi-metric multi-view stereo network. In *International Conference on Image Processing*, 2021. 1, 2

[19] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-to-end deep plane sweep stereo. In *International Conference on Learning Representations*, 2019. 2

[20] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisin Mac Aodha, Gabriel J. Brostow, and Jamie Watson. MVSAnywhere: Zero shot multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2025. 2

[21] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 1

[22] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[23] Mukul Khanna*, Yongsen Mao*, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. In *Conference on Computer Vision and Pattern Recognition*, 2024. 1

[24] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *arXiv preprint arXiv:1905.02706*, 2019. 1, 2

[25] Taekyung Kim, Jaehoon Choi, Seokeon Choi, Dongki Jung, and Changick Kim. Just a few points are all you need for multi-view stereo: A novel semi-supervised learning method for multi-view stereo. In *International Conference on Computer Vision*, 2021. 2

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[27] Jingliang Li, Zhengda Lu, Yiqun Wang, Ying Wang, and Jun Xiao. DS-MVSNet: Unsupervised multi-view stereo via depth synthesis. In *ACM International Conference on Multimedia*, 2022. 1, 2

[28] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[29] Jinli Liao, Yikang Ding, Yoli Shavit, Dihe Huang, Shihao Ren, Jia Guo, Wensen Feng, and Kai Zhang. WT-MVSNet: Window-based transformers for multi-view stereo. In *Advances in Neural Information Processing Systems*, 2022. 1, 2

[30] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *International Conference on Computer Vision*, 2019. 2

[31] Zhenxing Mi, Chang Di, and Dan Xu. Generalized binary search network for highly-efficient multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1

[32] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3D scene reconstruction from posed images. In *Conference on Computer Vision and Pattern Recognition*, 2020. 5

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 2019. 5

[34] Ke Qiu, Yawen Lai, Shiyi Liu, and Ronggang Wang. Self-supervised multi-view stereo via inter and intra network pseudo depth. In *International Conference on Multimedia*, 2022. 1, 2

[35] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 2, 3

[36] Alexander Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. 3DVNet: Multi-view depth prediction and volumetric refinement. In *International Conference on 3D Vision*, 2021. 1, 2

[37] Alex Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. Smoothness, synthesis, and sampling: Re-thinking unsupervised multi-view stereo with DIV loss. In *European Conference on Computer Vision*, 2024. 1, 2, 3, 4, 5, 7

[38] 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 *International Conference on Computer Vision*, 2021. 1, 2, 5, 7

[39] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2, 4

[41] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. SimpleRecon: 3D reconstruction without 3D convolutions. In *European Conference on Computer Vision*, 2022. 2, 4

[42] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2016. 1

[43] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large scale multi-view stereo for ultra high resolution image sets. *Machine Vision and Applications*, 23, 2011. 1

[44] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. SparseNeRF: Distilling depth ranking for few-shot novel view synthesis. In *International Conference on Computer Vision*, 2023. 2

[45] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision*, 2022. 2

[46] Zhou Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002. 4

[47] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems, & Computers*, 2003. 2, 3, 4

[48] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 4

[49] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. FoundationStereo: Zero-shot stereo matching. *Conference on Computer Vision and Pattern Recognition*, 2025. 2

[50] Cho-Ying Wu, Jialiang Wang, Michael Hall, Ulrich Neumann, and Shuochen Su. Toward practical monocular indoor depth estimation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[51] Junhua Xi, Yifei Shi, Yijie Wang, Yulan Guo, and Kai Xu. RayMVSNet: Learning ray-based 1D implicit fields for ac-

curate multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[52] Yuting Xiao, Jingwei Xu, Zehao Yu, and Shenghua Gao. DebSDF: Delving into the details and bias of neural indoor scene reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[53] Kaiqiang Xiong, Rui Peng, Zhe Zhang, Tianxing Feng, Jianbo Jiao, Feng Gao, and Ronggang Wang. CL-MVSNet: Unsupervised multi-view stereo with dual-level contrastive learning. In *International Conference on Computer Vision*, 2023. 1, 2, 5, 7

[54] Hongbin Xu, Zhipeng Zhou, Yu Qiao, Wenxiong Kang, and Qiuxia Wu. Self-supervised multi-view stereo via effective co-segmentation and data-augmentation. In *AAAI Conference on Artificial Intelligence*, 2021. 2, 5

[55] Hongbin Xu, Zhipeng Zhou, Yali Wang, Wenxiong Kang, Baigui Sun, Hao Li, and Yu Qiao. Digging into uncertainty in self-supervised multi-view stereo. In *International Conference on Computer Vision*, 2021. 1, 2

[56] Hongbin Xu, Weitao Chen, Yang Liu, Zhipeng Zhou, Haihong Xiao, Baigui Sun, Xuansong Xie, and Wenxiong Kang. Semi-supervised deep multi-view stereo. In *International Conference on Multimedia*, 2023. 2

[57] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1

[58] Jiayu Yang, Jose M. Alvarez, and Miaomiao Liu. Self-supervised learning of depth inference for multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2

[59] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Conference on Computer Vision and Pattern Recognition*, 2024. 2

[60] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 2

[61] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2018. 1, 2

[62] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *International Conference on Computer Vision*, 2023. 1, 2, 5, 7

[63] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In *European Conference on Computer Vision*, 2020. 2

[64] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems*, 2022. 2

[65] Weida Zhan, Keliang Cao, Yichun Jiang, Yu Chen, Jiale Wang, and Yang Hong. A Semi-Supervised Method for PatchMatch Multi-View Stereo with Sparse Points. *Photonics*, 9(12):983, 2022. 2

[66] Jinzhi Zhang, Ruofan Tang, Zheng Cao, Jing Xiao, Ruqi Huang, and Lu Fang. ElasticMVS: Learning elastic part representation for self-supervised multi-view stereopsis. In *Advances in Neural Information Processing Systems*, 2022. 1

[67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 4

[68] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 5