

Masked Depth Modeling for Spatial Perception

Bin Tan Changjiang Sun Xiage Qin Hanat Adai Zelin Fu Tianxiang Zhou
 Han Zhang Yinghao Xu Xing Zhu Yujun Shen Nan Xue[†]

[†]Project Lead

Spatial visual perception is a fundamental requirement in physical-world applications like autonomous driving and robotic manipulation, driven by the need to interact with 3D environments. Capturing pixel-aligned metric depth using RGB-D cameras would be the most viable way, yet it usually faces obstacles posed by hardware limitations and challenging imaging conditions, especially in the presence of specular or texture-less surfaces. In this work, we argue that the inaccuracies from depth sensors can be viewed as “masked” signals that inherently reflect underlying geometric ambiguities. Building on this motivation, we present LingBot-Depth, a depth completion model which leverages visual context to refine depth maps through *masked depth modeling* and incorporates an automated data curation pipeline for scalable training. It is encouraging to see that our model outperforms *top-tier RGB-D cameras* in terms of both depth precision and pixel coverage. Experimental results on a range of downstream tasks further suggest that LingBot-Depth offers an aligned latent representation across RGB and depth modalities. We release the code, checkpoint, and 3M RGB-depth pairs (including 2M real data and 1M simulated data) to the community of spatial perception.

Website: <https://technology.robbbyant.com/lingbot-depth>

Github: <https://github.com/robbbyant/lingbot-depth>

Checkpoints: <https://huggingface.co/robbbyant/lingbot-depth>



1 Introduction

Precise perception of the 3D world is fundamental to any intelligent agent operating in the physical environment—from biological organisms to autonomous vehicles and generalist robots. It provides the essential context for localization (“where am I?”) and scene understanding (“what is around me?”). Without robust 3D perception, real-world actions cannot be reliably planned, executed, or verified. Consequently, the pursuit of accurate 3D sensing has become a central pillar of research on physical-grounded artificial intelligence. The optimal approach to achieving this goal remains a subject of debate, with current methods generally falling into three paradigms: (1) classic multi-view visual geometry and recent learning-based adventures, (2) data-driven monocular depth estimation, and (3) active sensor-based depth measurement (*e.g.*, LiDAR, ToF, Structured Light). While the trade-offs of each paradigm have been extensively discussed [13, 29, 33], the requirements for effective perception came down to three critical criteria: (1) absolute metric scale, (2) pixel-aligned dense geometry, and (3) real-time acquisition without computationally expensive post-processing.

RGB-D cameras distinguish themselves as the only modality capable of satisfying these requirements in real-time. However, their utility is frequently compromised by inherent hardware limitations, particularly the susceptibility of stereo matching algorithms to appearance ambiguities. As illustrated in the left of Fig. 1, even state-of-the-art commercial sensors struggle in challenging scenarios—such as low-texture surfaces, specular reflections, and complex lighting conditions. These failures manifest as severe data corruption and missing values, directly violating the requirement for dense, pixel-aligned geometry.

In this work, we propose a paradigm shift: rather than treating these sensor failures as noise to be discarded, we leverage them as a useful learning signal. Inspired by recent advances in self-supervised masked modeling [9] and joint embedding architectures [1], we introduce **Masked Depth Modeling** (MDM) to achieve dense, pixel-aligned scene geometry. A key innovation of MDM is the interpretation of missing regions (“holes”) in raw depth maps as “natural

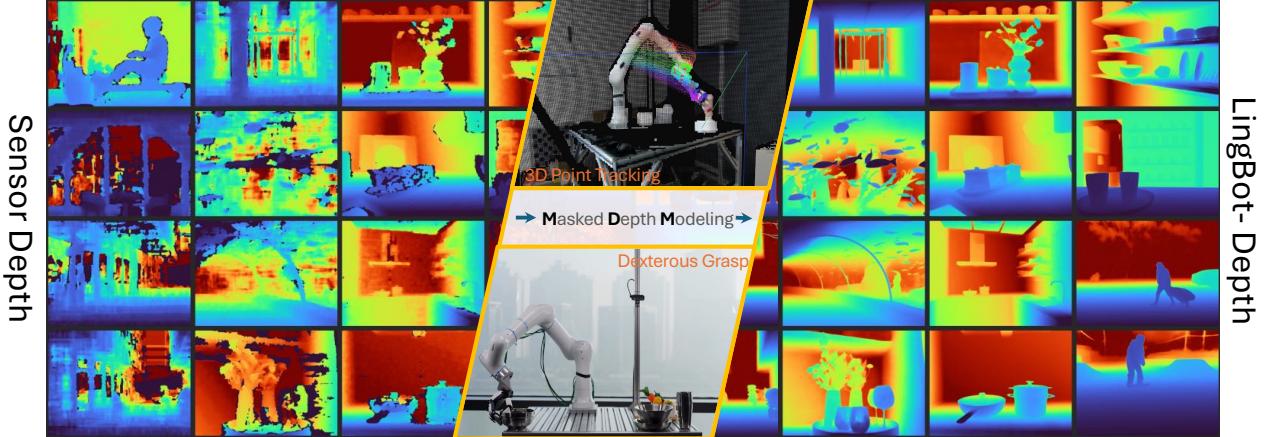


Figure 1. Enhanced sensor depth powered by our proposed MDM pretraining, which leverages naturally missing depth measurements in RGB-D sensors as masks to learn metric-scale, complete, and accurate depth representations. The resulting LingBot-Depth model serves as a powerful spatial perception prior for downstream applications, including 3D point tracking and dexterous grasping.

masks”, diverging from standard MAE approaches that rely solely on random masking. Because these natural masks arise specifically from geometric and appearance ambiguities (*e.g.*, specular reflections), they present a significantly harder reconstruction challenge than random dropout. To solve this, our architecture provides the full, unmasked RGB image as a condition. The model is thus forced to infer the missing depth values by reasoning about the correlation between the complete RGB context and the remaining valid depth tokens.

By unifying the objectives of monocular depth estimation and depth completion, our MDM framework serves as a versatile generalist capable of yielding metric-scale, pixel-aligned, dense depth maps from arbitrary RGB-D inputs. The transition between these tasks is governed simply by the masking strategy within the Transformer. In the extreme case where all depth tokens are masked, the model functions as a pure Monocular Depth Estimator, forcing the Self-Attention layers to exploit RGB context alone to infer geometry. Conversely, for Depth Completion, we mask only the invalid (sensor-corrupted) tokens, allowing the model to fuse the sparse valid depth readings with visual cues to reconstruct a complete, dense depth prediction.

To support large-scale MDM training, we built a scalable data curation pipeline that bridges raw sensor data and reliable supervision. The pipeline includes two parallel streams: a synthetic branch based on self-hosted 3D assets, and a real-world branch using a modular, 3D-printed capture rig compatible with a variety of consumer RGB-D cameras, including active stereo (Intel RealSense, Orbbec Gemini) and passive stereo (ZED) systems. Using this setup, we collected 1M synthetic samples and 2M real captures, each containing synchronized RGB images, raw sensor depth, and stereo pairs. The stereo pairs enable pseudo-depth supervision through a custom stereo matching network adapted from FoundationStereo [33] and trained on synthetic data. We further enrich this corpus with several public RGB-D datasets [4, 19, 21, 22, 30, 36, 37], forming a diverse training set for robust depth completion. Pretraining a ViT-Large on this curated RGB-D corpus with Masked Depth Modeling allows metric geometry to be integrated into semantic tokens via attention, thus improving the sensing quality of RGB-D cameras, as illustrated in the right panel of Fig. 1.

Experimentally, we first validate MDM on standard benchmarks, where it achieves competitive performance in both depth completion and metric monocular depth estimation. Moreover, it serves as a stronger monocular depth prior for FoundationStereo [33] compared to DepthAnythingV2 [35]. Despite being trained solely on static images, our model demonstrates remarkable zero-shot generalization to video depth estimation, producing temporally consistent geometry without explicit temporal supervision. Leveraging this spatiotemporal consistency, we deploy our pretrained model as a drop-in depth estimator for 3D tracking, replacing the standard VGGT [27] frontend in SpatialTrackerV2 [34]. This leads to improved motion understanding and higher computational efficiency in real-world scenarios. Finally, we demonstrate practical robotic utility by training a dexterous grasping policy. Unlike prior approaches that depend on object CAD models or perfect simulator states, our policy leverages the robust depth predictions and latent representations learned through MDM to enable open-world grasping. We successfully grasp challenging objects that typically defeat conventional sensors, including transparent glassware and highly reflective bowls.

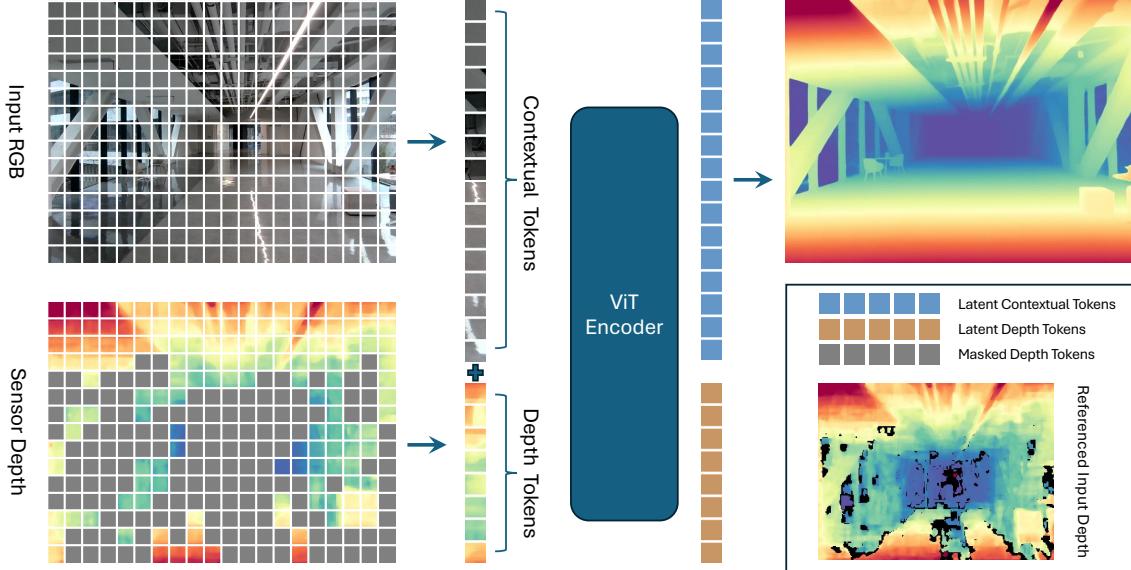


Figure 2. Illustration of the proposed Masked Depth Modeling framework. Depth tokens corresponding to missing sensor measurements are masked, and a ViT encoder learns a joint embedding from the contextual tokens (i.e., the RGB frame) and the remaining unmasked depth tokens. In the decoder stage, latent depth tokens are discarded, and a ConvStack decoder reconstructs the full depth map from latent contextual tokens. We put an unmasked depth map in the bottom-right as the reference.

2 Masked Depth Modeling

Masked depth modeling follows the general paradigm of masked image modeling [9] within an encoder–decoder framework, but shifts the learning objective from appearance reconstruction to depth map prediction by operating on RGB-D inputs. We adopt a standard Vision Transformer (ViT) architecture [5], using ViT-Large as the encoder backbone. For depth reconstruction from patch tokens, instead of the shallow Transformer decoder used in vanilla MAE [9], we employ a ConvStack decoder adopted from MoGe [28, 29], which is better suited for dense geometric prediction. Figure 2 provides an overview of the masked depth modeling architecture.

2.1 Separated Patch Embedding for RGB-D Inputs

Patch Embedding Layers. We apply separate patch embedding layers to the two input modalities: the 3-channel RGB image and the single-channel depth map. Each modality is independently projected into a sequence of patch tokens, yielding RGB tokens and depth tokens that are spatially aligned on the same 2D grid. This separated patch embedding design enables the self-attention layers in the ViT encoder to learn joint representations that integrate appearance context from RGB images with geometric cues from depth measurements. In particular, the model can exploit complementary information from rich visual context and fundamental geometric priors such as near–far relationships, coplanarity, and spatial continuity. In our implementation, we set the patch size to 14 for both patch embedding layers, following DINOv2 [18]. Without loss of generality, we assume that the RGB frame and the depth map share the same spatial resolution (H, W) , where both H and W are divisible by 14. The number of tokens per modality is therefore $N = HW/14^2$. We denote the i -th RGB token as $\mathbf{c}_i \in \mathbb{R}^n$ and the i -th depth token as $\mathbf{d}_i \in \mathbb{R}^n$, where n is the token embedding dimension.

Positional Embeddings. Compared to standard ViT inputs, RGB-D token sequences require encoding two types of positional information: (1) the 2D spatial location of each token, and (2) the modality identity, distinguishing RGB tokens from depth tokens at the same spatial position. To this end, we introduce two types of positional embeddings: (1) a shared learnable 2D spatial positional embedding for both RGB and depth tokens, capturing their spatial locations in the image plane; and (2) a modality embedding that distinguishes the input source of each token—RGB or depth. Specifically, the modality embedding is set to 1 for RGB tokens and 2 for depth tokens. The final positional encoding of each token is computed as the sum of its spatial and modality embeddings. Finally, each RGB/depth token \mathbf{c}_i or \mathbf{d}_i are added with the corresponding positional embedding before feeding to the Attention blocks.

2.2 Joint Embedding of RGB and Unmasked Depth for Masked Depth Prediction

Depth sensors are inherently sensitive to appearance-related disturbances, and it is common for RGB-D cameras to produce depth maps with missing or invalid measurements. Importantly, such missing depth values often correlate with challenging visual conditions in the scene, including surface material properties, lighting variations, and reflective or textureless regions. Rather than treating these missing measurements as noise, we view them as an opportunity to learn a joint representation from the complementary sources: valid depth observations and the always-available RGB appearance context. Accordingly, we leverage the masking patterns naturally induced by depth sensors and train the model to learn a joint embedding of RGB tokens and unmasked depth tokens, enabling robust depth reasoning under incomplete observations.

Masking from Missing Depth Measurements. Our masking strategy is grounded in the inductive bias introduced by missing depth measurements. In practice, however, many RGB-D samples are well conditioned and contain few or no missing depth values. Such samples are nonetheless valuable for learning joint embeddings, as they provide abundant paired appearance–geometry observations and should not be discarded. Accordingly, we aim to incorporate as many RGB-D samples as possible to support large-scale training of Vision Transformers. Another practical consideration arises from patch-based processing: a single patch may contain a mixture of valid and invalid depth values. To address this ambiguity, we define the masking decision at the patch level based on the validity statistics within each patch, ensuring a consistent and well-defined masking rule for depth tokens.

Based on the above discussion, a depth patch (token) whose values are entirely missing is always masked. For patches containing a mixture of valid and invalid depth values, we assign a higher probability (*e.g.*, 0.75 in our training) of being masked. If the number of masked tokens from these two cases is insufficient to meet the target masking ratio, we randomly sample additional fully valid depth tokens to complete the masking set. Such a masking strategy allows imperfect yet informative depth tokens to remain unmasked, enabling meaningful interaction with contextual RGB tokens. The overall masking ratio is in the range of 60%–90% for depth maps.

RGB-D Tokens for Vision Transformers. After applying the masking strategy, the full set of RGB tokens and the unmasked depth tokens are concatenated to form RGB-D tokens, which are fed into a ViT-Large encoder with 24 self-attention blocks. In addition to the RGB-D tokens, a [cls] token is retained to capture global context across modalities. Different from conventional designs that aggregate tokens from multiple intermediate layers (*e.g.*, layers 6, 12, 18, and 24) for dense prediction, as adopted in DepthAnythingV2 [35] and MoGe [28, 29], we only retain the output tokens from the final encoder layer for subsequent processing.

In vanilla MAE, pixel-wise RGB reconstruction is performed using a shallow Transformer decoder. Each patch token is decoded into $3 \times \text{patch_size} \times \text{patch_size}$ values, which are then reshaped into the pixel domain. This design is suitable for RGB-only masking, where masked tokens share homogeneous latent representations within the decoder. In contrast, our masking strategy is applied exclusively to depth tokens, while the RGB tokens remain fully observable and provide complete contextual information. This setting allows depth prediction at each spatial location to be conditioned on rich appearance context. Consequently, we adopt a convolutional decoder architecture, termed *ConvStack*, which is more appropriate for dense geometric reconstruction.

ConvStack Decoder. After the encoder, latent depth tokens are discarded, while the latent contextual tokens are retained as spatially distributed representations. To inject global scene context, the [cls] token is broadcast and element-wise added to each contextual token, enriching them with task-level semantic information. These enhanced tokens serve as input queries to a hierarchical convolutional decoder adapted from MoGe [28, 29].

The decoder follows a pyramid structure composed of a shared convolutional neck and multiple task-specific heads. Starting from a low-resolution feature map of size (h, w) , the neck progressively upsamples features through stacked residual blocks and transposed convolutions (kernel size 2, stride 2), doubling the spatial resolution at each stage until reaching $(16h, 16w)$. At each scale, UV positional encodings—derived from a circular mapping of image coordinates—are injected to preserve spatial layout and aspect ratio. The resulting multi-scale feature pyramid is shared across all task heads, enabling efficient feature reuse while allowing each head to decode its own dense output. The final depth prediction is bilinearly upsampled to match the original input resolution. This design decouples high-level context modeling from dense geometric reconstruction, enabling precise depth estimation while supporting scalable and flexible multi-task learning.

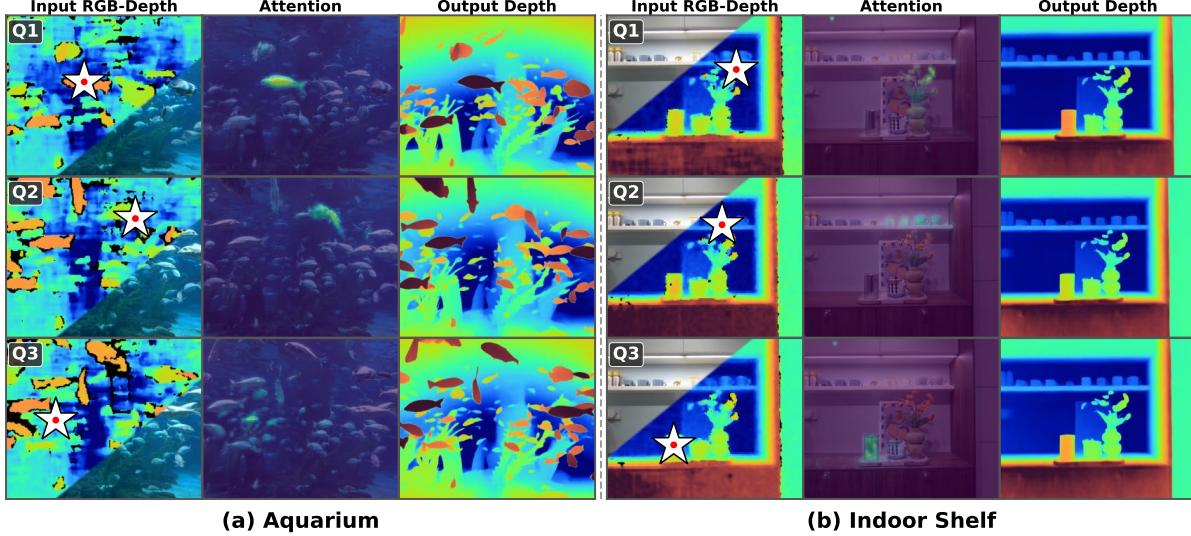


Figure 3. Multi-query depth-to-RGB attention visualization. For two scenes, (a) an aquarium with densely packed objects and (b) an indoor shelf with heterogeneous materials, we select three depth query patches (Q1–Q3) and visualize their attention over RGB tokens. Each row shows the masked input depth (with query location marked by \star), the attention overlay on the RGB image, and the refined depth output. Different queries attend to distinct, spatially corresponding regions, confirming that the joint embedding captures fine-grained cross-modal geometric–appearance associations. The RGB-D camera we used here is Orbbee Gemini-335.

Attention Visualization. To verify that the joint embedding effectively captures cross-modal associations, we visualize the depth-to-RGB attention maps from the final encoder layer. For a selected depth token, we extract its attention weights over all RGB tokens and project them back onto the image plane as a heatmap overlay. As shown in Fig. 3, we select multiple query patches within the same scene and visualize their respective attention patterns. Across diverse scenarios—an aquarium with densely packed objects at varying depths, and an indoor shelf with heterogeneous materials—different depth tokens consistently attend to distinct, spatially localized regions in the RGB image that correspond to their respective query locations. This demonstrates that the encoder learns fine-grained, position-aware geometric–appearance correspondences through the masked depth modeling objective, rather than collapsing to global or trivial attention patterns.

2.3 Training Details

We adopt a 24-layer ViT-Large encoder (ViT-L/14) [18] as the visual backbone. Since training Vision Transformers is data-intensive and our goal is to explore joint contextual modeling of RGB frames and depth maps, we initialize the encoder with the official DINOv2 pretrained checkpoint. The decoder, which consists of a shared convolutional neck and task-specific heads, is randomly initialized. To accommodate the different optimization dynamics of the pretrained encoder and the randomly initialized decoder, we employ a differential learning rate strategy: parameters in the encoder backbone are optimized with a base learning rate of 1×10^{-5} , while all remaining parameters, including those in the decoder, are trained with a higher learning rate of 1×10^{-4} . The optimizer is AdamW [15] with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.05. Parameter groups are defined based on module names, such that all parameters matching the pattern `*backbone*` are assigned to the low-learning-rate group. We employ a composite learning rate schedule. During the first 2,000 iterations, the encoder learning rate is linearly warmed up from zero to 1×10^{-5} , while the decoder learning rate starts directly at its target value. After warm-up, a step decay scheduler reduces both learning rates by a factor of 0.5 every 25,000 iterations.

Training is conducted for a total of 250,000 iterations with a global batch size of 1,024, achieved using 128 GPUs and a per-GPU batch size of 8. Data augmentation includes random resized cropping and horizontal flipping, along with a set of synthetic image degradations—specifically color jittering, JPEG compression artifacts, motion blur, and shot noise—to improve robustness under realistic visual conditions. Gradient clipping with a maximum norm of 1.0 is applied to stabilize optimization, and mixed-precision training using BF16 is enabled throughout to improve computational efficiency and reduce memory consumption. The entire training process takes approximately 7.5 days. The predicted

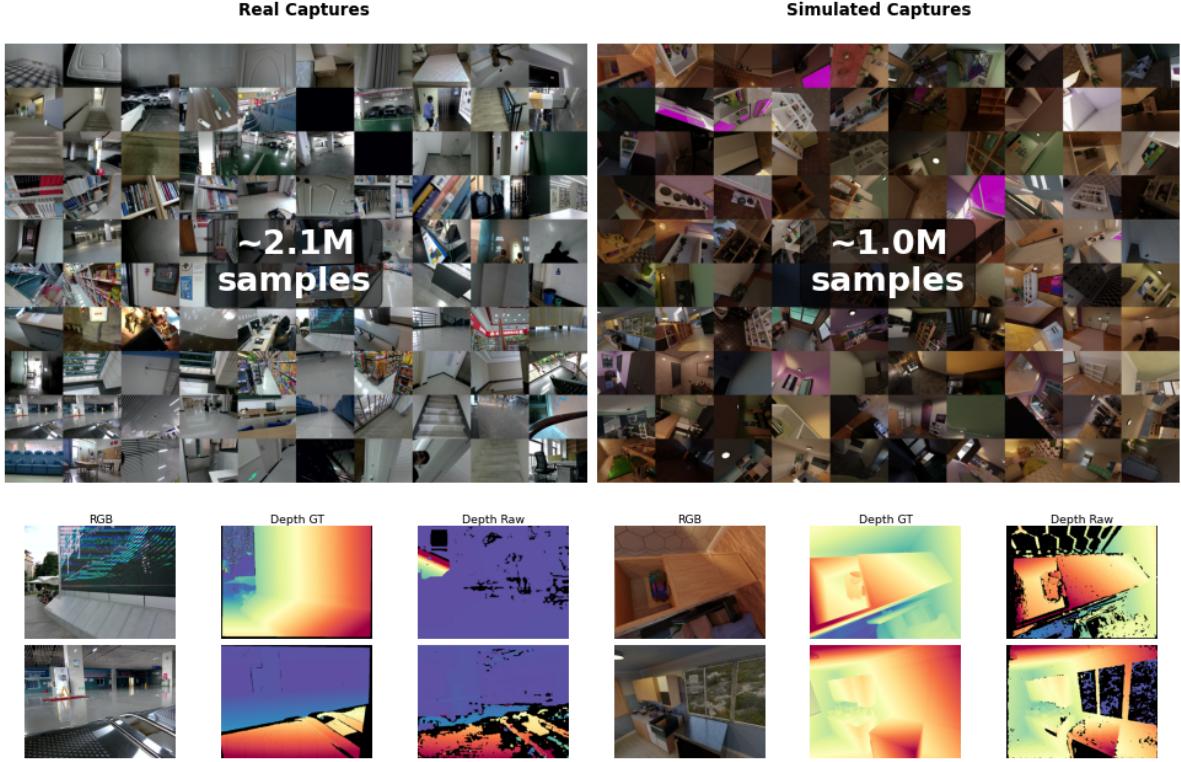


Figure 4. Our data curation pipelines. Samples from a total of 2.1M real-captured samples plus 1.0M simulated captures are gathered in the top row. In the bottom row, we show the RGB-D inputs and the GT depth maps accordingly.

depth map is supervised using an L1 loss applied directly to the ground-truth depth map. Only pixels with valid depth values in the ground truth are included in the loss computation.

3 Data Curation Pipelines

RGB-D data is considerably scarcer than RGB-only data due to the reliance on specialized depth sensors. Most existing RGB-D datasets either avoid challenging imaging conditions to reduce missing depth measurements or generate near-perfect depth maps using high-quality 3D assets and rendering engines. Consequently, they lack the naturally occurring depth incompleteness required for masked depth modeling. To overcome this limitation, we curate RGB-D data that preserves realistic missing patterns induced by real-world sensing. Our data curation pipeline consists of two parallel streams: a synthetic pipeline built on self-hosted 3D assets, and a real-world pipeline based on a scalable RGB-D capture system using multiple commercial depth cameras. The mask ratio distributions in the raw data of our data is shown in Fig. 5. We name the synthetic data curated from our synthetic pipeline as LingBot-Depth-S, and the real-world data curated from the real-world pipeline as LingBot-Depth-R. In addition, we incorporate existing open-source RGB-D datasets as supplementary training data. For these datasets, we artificially corrupt depth maps with Gaussian noise and apply our masking strategy during training, while using the original depth maps as reconstruction targets with valid-pixel masks to exclude missing measurements.

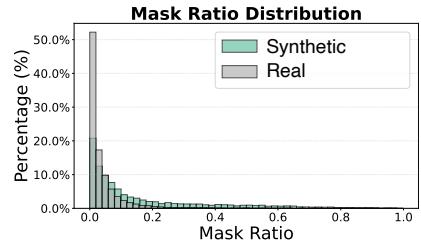


Figure 5. Mask ratio distributions for our curated synthetic and real-world RGB-D datasets. We compute the ratio of invalid depth pixels as the original mask ratio. Note that the synthetic data was processed by open-source SGM [10] algorithm, it has more missing measurements in the simulated sensor depth than the real captures.

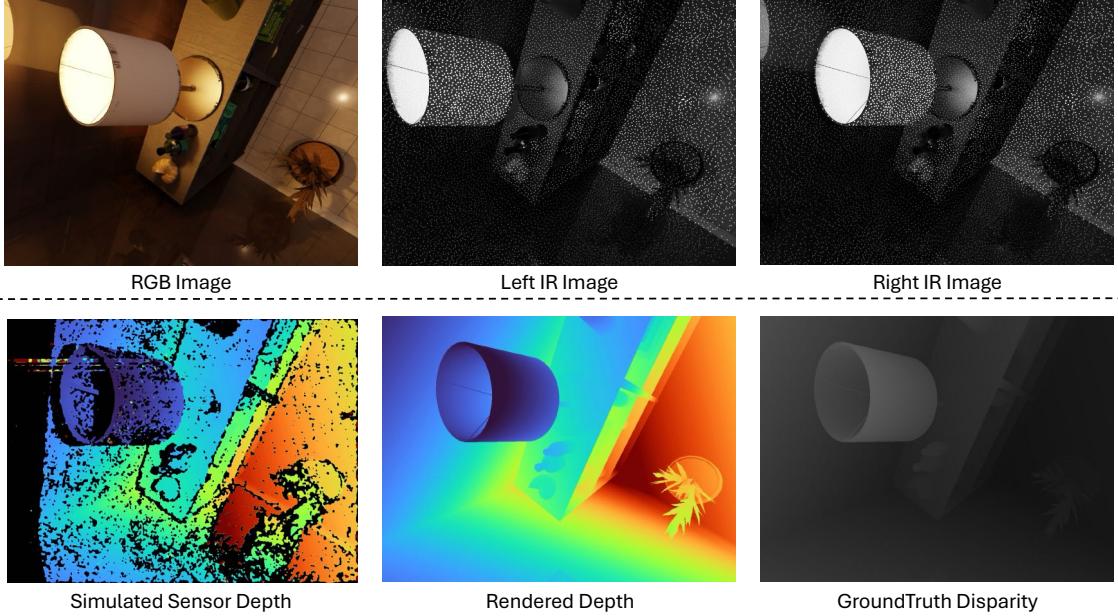


Figure 6. A synthetic data sample from our pipeline. Each sample includes an RGB image, a perfect rendered depth, a stereo pair with speckle patterns, a ground-truth disparity map, and a simulated sensor depth computed via semi-global matching (SGM) to mimic real-world active depth camera artifacts.

3.1 Synthetic Data Pipeline

Our synthetic data pipeline differs from existing approaches that focus solely on rendering idealized depth maps. Instead, we explicitly simulate the imaging process of real-world active RGB-D cameras to generate realistic depth observations with natural imperfections.

Using self-hosted 3D assets, we simultaneously render RGB images, perfect depth maps, and grayscale stereo image pairs with speckle patterns in Blender. The RGB image is rendered from the left camera of the stereo pair, ensuring pixel-wise alignment between RGB appearance and the depth measurements produced by stereo matching. The rendered stereo images are then processed using the widely adopted semi-global matching (SGM) algorithm to produce sensor-like depth maps that mimic real-world capture artifacts.

For stereo rendering, we configure a pair of virtual cameras in Blender and randomly sample the stereo baseline from a uniform distribution between 0.05 and 0.2 meters. The camera focal length is independently sampled from a uniform distribution between 16 and 28 millimeters, enabling diverse imaging geometries.

Resulting Data Sample Format. We render 1 million synthetic samples from 442 indoor scenes. Each sample (as shown in Fig. 6) contains the following components:

- An **RGB image** rendered at a resolution of 960×1280 ($H \times W$);
- A **perfect depth map** rendered at the same target resolution;
- A **stereo image pair** rendered at a resolution of 720×960 and the groundtruth **disparity map** rendered at 960×1280 resolution;
- A **sensor depth map** computed from the stereo pair at 720×960 and then upsampled using nearest-neighbor interpolation to match the target resolution.

Comparisons to Existing Datasets. Prior to our work, several efforts have explored the simulation of imperfect depth measurements [3, 32], but at a substantially smaller scale. For example, HSSD-IsaacSIM-STD [32] renders approximately 10k stereo infrared image pairs, while DREDS [3] generates about 130k stereo pairs. In contrast, our simulation pipeline produces data at a scale more than an order of magnitude larger. Beyond data quantity, our

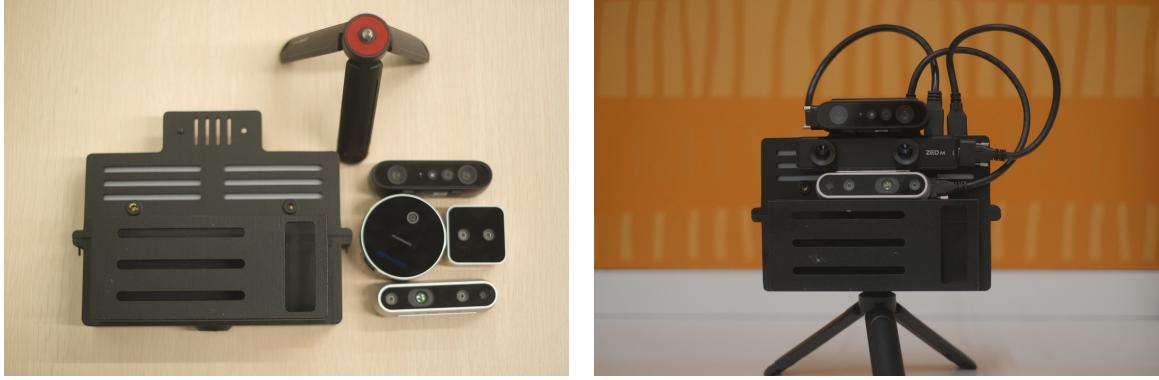


Figure 7. Our scalable RGB-D capture system. Left: individual hardware components, including multiple RGB-D sensors, mounting fixtures, and supporting accessories. Right: the assembled capture system mounted on a tripod, enabling scalable and flexible real-world RGB-D data acquisition.

Table 1. Distribution of our real-world RGB-D captures by scene category. We design a diverse collection protocol spanning residential, commercial, public, and specialized spaces to ensure broad coverage of indoor environments.

| Residential Space | | Work / Study Space | | Commercial / Service | |
|-----------------------|--------|---------------------|-------|----------------------|--------|
| < 50 m ² | 10.16% | Office | 3.39% | Retail Store | 3.39% |
| 50–100 m ² | 10.16% | Meeting Room | 3.39% | Restaurant / Café | 3.39% |
| > 100 m ² | 10.16% | Classroom | 3.39% | Hotel / Guestroom | 1.70% |
| | | Library | 1.70% | Gym | 3.39% |
| | | Workshop | 1.70% | Lobby / Corridor | 1.70% |
| | | Laboratory | 1.70% | | |
| Public Space | | Special-Function | | Outdoor Scene | |
| Hospital / Clinic | 3.39% | Parking Garage | 3.39% | Outdoor Environment | 10.16% |
| School / Hall | 3.39% | Machinery Room | 3.39% | | |
| Airport / Station | 3.39% | Elevator | 3.39% | | |
| Museum | 3.39% | Corridor / Passage | 3.39% | | |
| | | Storage / Warehouse | 3.39% | | |

dataset also differs qualitatively in terms of scene fidelity. We employ self-hosted, high-quality 3D scenes, rather than constructing scenes by simply stacking isolated objects in simulators, as done in DREDS [3]. Although HSSD-IsaacSIM-STD [32] adopts more realistic scene-level layouts, its reliance on robotics-oriented simulators introduces approximations that lead to reduced visual fidelity compared to real-world captures.

3.2 Scalable RGB-D Capture Systems

To collect large-scale real-world RGB-D data, we build a scalable RGB-D capture prototype, as shown in Fig. 7. Specifically, we design and fabricate custom mounting fixtures using 3D printing, which allows flexible attachment of different types of commercial RGB-D cameras to the rear side of the fixture. On the front side, a portable PC client equipped with a touchscreen is integrated to receive and manage the data streams from the cameras. We develop a unified data acquisition interface using the official SDKs provided by the respective camera manufacturers. This lightweight and modular design makes the capture system both scalable and user-friendly, significantly lowering the barrier for large-scale RGB-D data collection. We deploy multiple capture devices to scale up data acquisition over the scenes listed in Tab. 1. Because the real captures do not have missing-free depth maps, we follow FoundationStereo [33] to compute the stereo disparity of the left-right IR pairs to obtain the pseudo depth labels. To ensure the quality of pseudo depth maps from stereo matching, we do the left-right check and filter out the inconsistent pixel values from the depth maps. Finally, a total of 2M real captures with extraordinary scene diversity was obtained for masked depth modeling.

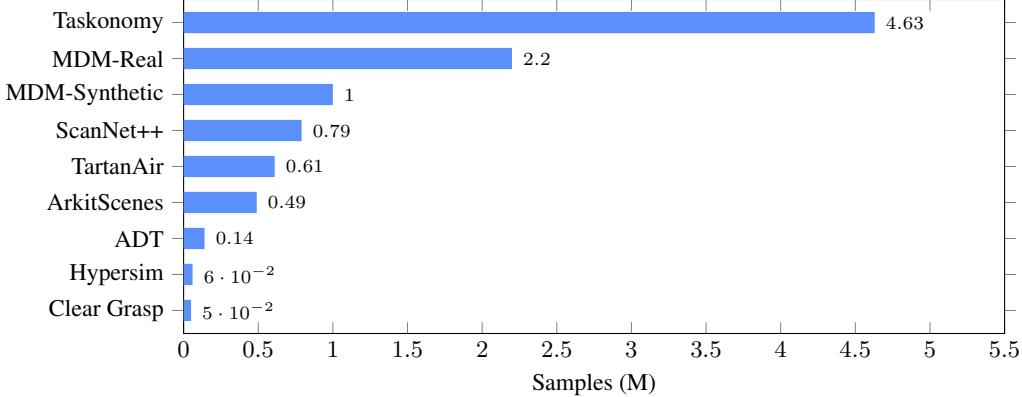


Figure 8. Data composition for MDM pretraining (~ 10 M samples total). Our self-curated data (MDM-Real and MDM-Synthetic, 3.2M) is combined with seven open-source RGB-D datasets to ensure diverse scene coverage and depth characteristics.

3.3 Training Data Summary

Besides of the self-curated 3.2M data (including the LingBot-Depth-S and LingBot-Depth-R splits), we also use the following open-source datasets [4, 19, 21, 22, 30, 36, 37] as training data to form a total of 10M training samples for masked depth modeling. Note that the open-source synthetic datasets have no missing depth measurements, we randomly generate the patchified tokens to meet the expected masking ratio range of 60%-90% without any additional processing. For the real-world open-source datasets, the masking strategy would also be dominated by the random mask sampling because there depth maps are relative complete than our curated data. A summary of the training data is shown in Fig. 8.

4 Experiments

In this section, we experimentally validate the effectiveness of the proposed MDM pretraining. First, we evaluate the pretrained MDM model, LingBot-Depth, on the task of depth completion (DC) in Sec. 4.1, which is most closely aligned with our pretraining objective. Second, we compare different pretrained backbones for monocular depth estimation by training MoGe [28] models using either DINOv2 or our MDM-pretrained weights. Third, we demonstrate that using our MDM-pretrained model as a foundational prior improves FoundationStereo compared to its original backbone, DepthAnythingV2 [35]. Building upon these results, in Sec. 5, we showcase three downstream applications: video depth completion, online 3D point tracking, and grasp pose generation for robotic dexterous manipulation.

4.1 Depth Completion

As MDM pretraining is naturally aligned with the depth completion task, we evaluate our pretrained LingBot-Depth model against state-of-the-art approaches, including OMNI-DC [39], PromptDA [13], and PriorDA [31]. The evaluation checkpoints for these methods are obtained from their official repositories. We follow two evaluation protocols, (1) Block-wise Depth Masking and (2) Sparse SfM Depth Inputs.

Protocol 1: Block-wise Depth Masking. We generate incomplete depth maps by randomly masking out spatial regions of varying sizes (blocks) from the ground-truth depth, simulating depth dropout commonly observed in consumer depth cameras. To further model low-quality depth measurements, we corrupt the remaining depth values with additive Gaussian noise and shot-noise-like perturbations inspired by Kinect noise models [17], which capture sensor-specific artifacts such as quantization effects and photon noise. Based on the severity of masking and noise level, each dataset is divided into four difficulty levels: *easy*, *medium*, *hard*, and *extreme*. As shown in Tab. 2a, our method consistently outperforms all baseline approaches across all difficulty levels, demonstrating strong robustness to both structural incompleteness and measurement noise. Under this protocol, three benchmark datasets, iBims [12], NYUv2 [25], and DIODE [26] are used for evaluation.

Table 2. Depth completion results. (a) Block-wise depth masking with four difficulty levels on iBims, NYUv2, and DIODE. (b) Sparse SfM depth inputs on ETH3D.

(a). Protocol 1: Block-wise Depth Masking.

| | Easy | | | | Medium | | | | Hard | | | | Extreme | | | | Easy | | | | Medium | | | | Hard | | | | Extreme | | | | |
|-----------------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-------|-------|---------------|---------|-------|-------|-------|-----------------|-------|-------|-------|--------|-------|-------|---------------|-------|-------|-------|-------|---------|-------|-------|-------|-------|
| | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | RMSE↓ | REL↓ | | | | | |
| iBims | | | | | | | | | | | | NYUv2 | | | | | | | | | | | | DIODE-Outdoor | | | | | | | | | |
| OMNI-DC [39] | 0.476 | 0.084 | 0.929 | 0.232 | 1.382 | 0.342 | 2.053 | 0.555 | 0.234 | 0.055 | 0.308 | 0.079 | 0.395 | 0.103 | 0.643 | 0.176 | 0.409 | 0.095 | 0.433 | 0.093 | 0.520 | 0.094 | 0.845 | 0.150 | 0.289 | 0.093 | 0.276 | 0.080 | 0.267 | 0.072 | 0.309 | 0.076 | |
| OMNI-DC-DA [39] | 0.602 | 0.068 | 0.771 | 0.155 | 1.266 | 0.302 | 1.937 | 0.538 | 0.343 | 0.053 | 0.385 | 0.070 | 0.418 | 0.088 | 0.456 | 0.116 | PromptDA [13] | 0.298 | 0.055 | 0.337 | 0.065 | 0.428 | 0.083 | 0.607 | 0.129 | 0.209 | 0.051 | 0.219 | 0.053 | 0.238 | 0.057 | 0.324 | 0.074 |
| PriorDA [31] | 0.409 | 0.095 | 0.433 | 0.093 | 0.520 | 0.094 | 0.845 | 0.150 | 0.289 | 0.093 | 0.276 | 0.080 | 0.267 | 0.072 | 0.309 | 0.076 | Ours | 0.175 | 0.020 | 0.223 | 0.038 | 0.289 | 0.060 | 0.345 | 0.083 | 0.089 | 0.014 | 0.124 | 0.021 | 0.146 | 0.027 | 0.181 | 0.033 |
| DIODE-Indoor | | | | | | | | | | | | DIODE-Outdoor | | | | | | | | | | | | DIODE-Outdoor | | | | | | | | | |
| OMNI-DC [39] | 0.563 | 0.074 | 0.895 | 0.121 | 1.108 | 0.153 | 1.775 | 0.271 | 2.214 | 0.041 | 2.806 | 0.067 | 3.701 | 0.103 | 6.239 | 0.204 | OMNI-DC-DA [39] | 0.710 | 0.065 | 0.991 | 0.097 | 1.222 | 0.121 | 1.592 | 0.211 | 2.229 | 0.053 | 3.703 | 0.070 | 5.352 | 0.110 | 5.988 | 0.173 |
| PromptDA [13] | 0.250 | 0.049 | 0.291 | 0.054 | 0.320 | 0.060 | 0.465 | 0.083 | 2.587 | 0.070 | 2.807 | 0.087 | 3.026 | 0.100 | 4.313 | 0.156 | PriorDA [31] | 0.370 | 0.073 | 0.354 | 0.064 | 0.369 | 0.062 | 0.665 | 0.083 | 2.734 | 0.090 | 2.693 | 0.085 | 2.909 | 0.088 | 5.114 | 0.136 |
| Ours | 0.090 | 0.013 | 0.185 | 0.028 | 0.195 | 0.031 | 0.221 | 0.032 | 2.011 | 0.049 | 2.338 | 0.061 | 2.578 | 0.066 | 3.811 | 0.092 | | | | | | | | | | | | | | | | | |

(b). Protocol 2: Sparse SfM Depth Inputs (ETH3D).

| | Indoor | | | | Outdoor | | | |
|-----------------|--------|-------|-------|---------------------|---------|-------|-------|---------------------|
| | RMSE↓ | MAE↓ | REL↓ | $\delta_1 \uparrow$ | RMSE↓ | MAE↓ | REL↓ | $\delta_1 \uparrow$ |
| OMNI-DC [39] | 0.605 | 0.239 | 0.090 | 0.932 | 1.069 | 0.312 | 0.053 | 0.954 |
| OMNI-DC-DA [39] | 0.489 | 0.164 | 0.061 | 0.967 | 1.093 | 0.215 | 0.035 | 0.979 |
| PromptDA [13] | 1.023 | 0.648 | 0.242 | 0.796 | 2.750 | 1.612 | 0.300 | 0.776 |
| PriorDA [31] | 0.360 | 0.157 | 0.058 | 0.963 | 1.238 | 0.459 | 0.090 | 0.952 |
| Ours | 0.192 | 0.081 | 0.023 | 0.982 | 0.664 | 0.223 | 0.033 | 0.976 |

Protocol 2: Sparse SfM Depth Inputs. Using SfM/SLAM techniques to recover camera poses and sparse scene geometry is a common choice when the depth cameras are unavailable. Therefore, recovering the complete depth from the sparse SfM/SLAM points is valuable to many downstream applications, but yet being challenging because of the sparsity. This is a much more challenging protocol than the protocol 1. Following OMNI-DC [39], we apply the Sparse SfM observations on the ETH-SfM dataset [24] to evaluate the performance of our MDM-pretrained model.

Results. As shown in Tab. 2a, our method consistently outperforms all competing approaches across all four difficulty levels on every dataset under Protocol 1. On indoor benchmarks (iBims, NYUv2, DIODE-Indoor), LingBot-Depth reduces RMSE by over 40% relative to the best competitor (PromptDA) even in the *extreme* setting, demonstrating strong resilience to heavy masking and noise corruption. On DIODE-Outdoor, where depth ranges are significantly larger, our method still achieves the lowest errors across all levels. Under the more challenging Protocol 2 (see Tab. 2b), where only highly sparse SfM point clouds are available as input, LingBot-Depth again achieves state-of-the-art results on both indoor and outdoor splits of ETH-SfM, reducing RMSE by 47% (indoor) and 38% (outdoor) compared to the best baseline. Qualitative comparisons in Fig. 9 further illustrate that LingBot-Depth produces sharper depth boundaries and more coherent structures than competing methods, particularly in regions with severe occlusion or sparse observations. These results confirm that the MDM pretraining objective equips the model with strong depth priors that generalize across diverse corruption patterns and sparsity levels.

4.2 Monocular Depth Estimation

Monocular depth estimation has benefited significantly from large-scale visual pretraining [18]. As a new pretraining paradigm that learns joint representations of RGB appearance and depth measurements, we use our LingBot-Depth model as an alternative to DINOv2 [18] for initializing MoGe [28]. To adapt our pretrained model for monocular input, we remove the depth embedding branch and the ConvStack decoder, retaining only the encoder backbone for RGB images.

Training. We follow the standard fine-tuning protocols, and train two MoGe models (one with DINOv2 pretrain, another one with LingBot-Depth pretrain). To save the training time, we only use the TartanAir dataset [30] in the training instead of the official implementation of MoGe [28] that use much more datasets to boost the performance.

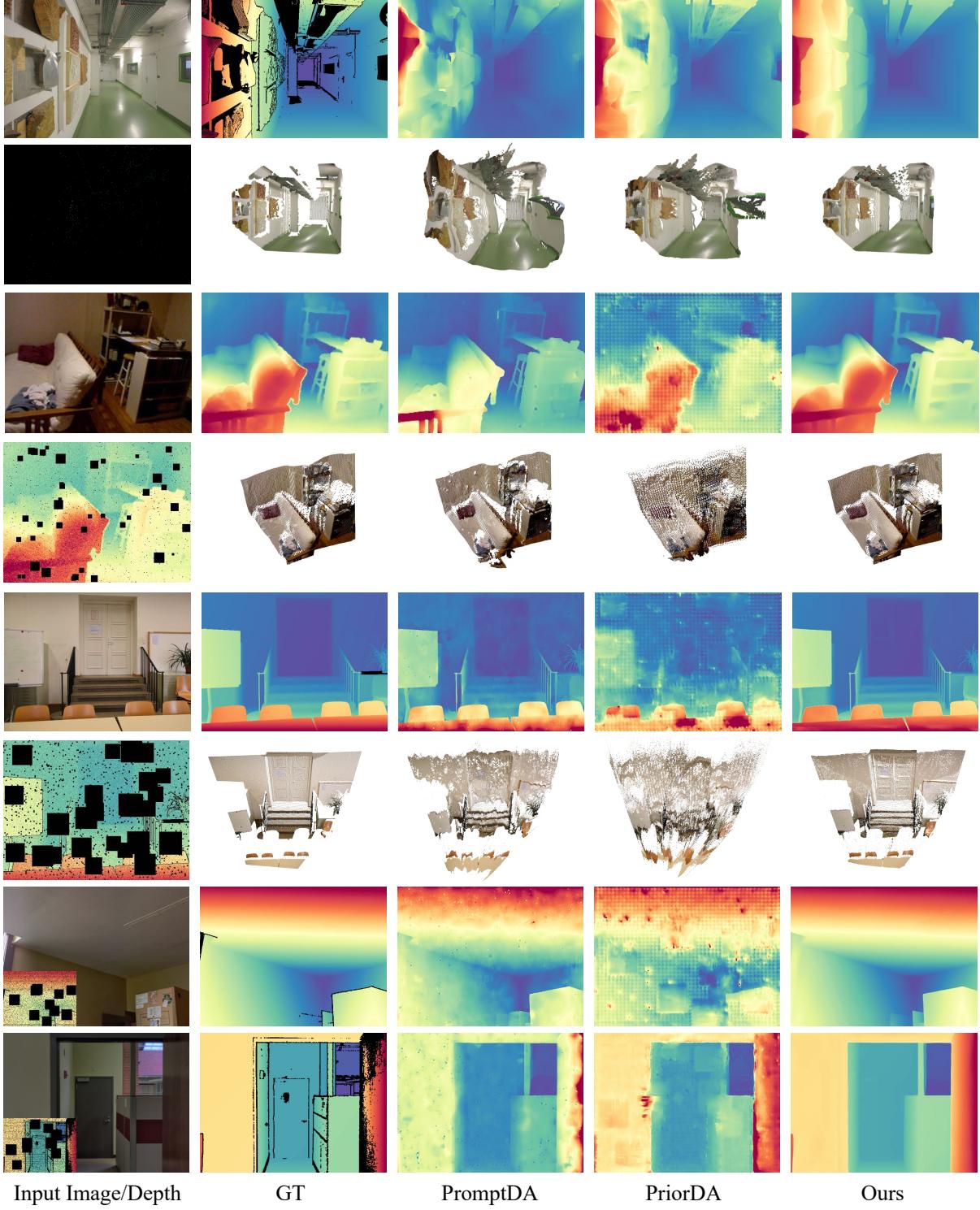


Figure 9. Qualitative comparison of depth completion results across four benchmarks. For each dataset, we show the RGB input, sparse/masked depth input, and predictions from OMNI-DC, PromptDA, PriorDA, and ours. Our method produces sharper boundaries and more complete structures, particularly in regions with severe occlusion or sparse observations.

Evaluation Protocol. The trained models are then evaluated on a wide range of diverse benchmarks, including indoor (NYUv2 [25], HAMMER [11]), outdoor driving (KITTI [7], DDAD [8]), large-scale synthetic (GSO [6], Sintel [2]), and challenging real-world scenes (iBims-1 [12], DIODE [26], Spring [16]).

Table 3. Monocular depth estimation results using MoGe with different pretrained backbones (DINOv2 vs. ours). We evaluate depth and point map accuracy under affine-invariant, scale-invariant, and disparity-invariant metrics across 10 diverse benchmarks.

| | Depth | | | | | | Point | | | | | | Depth | | | | | | Point | | | | | | |
|--------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|---------|
| | Aff-inv | | Scl-inv | | Dsp-inv | | Aff-inv | | Scl-inv | | Aff-inv | | Scl-inv | | Dsp-inv | | Aff-inv | | Scl-inv | | Dsp-inv | | Aff-inv | | Scl-inv |
| | REL \downarrow | $\delta_1 \uparrow$ | |
| NYUv2 | | | | | | | | | | | | KITTI | | | | | | | | | | | | | |
| DINOv2 | 0.056 | 0.965 | 0.068 | 0.948 | 0.061 | 0.966 | 0.066 | 0.957 | 0.068 | 0.948 | 0.071 | 0.934 | 0.226 | 0.544 | 0.154 | 0.835 | 0.204 | 0.615 | 0.228 | 0.539 | | | | | |
| Ours | 0.044 | 0.976 | 0.054 | 0.966 | 0.049 | 0.978 | 0.052 | 0.971 | 0.054 | 0.966 | 0.066 | 0.936 | 0.220 | 0.556 | 0.144 | 0.859 | 0.208 | 0.599 | 0.222 | 0.549 | | | | | |
| ETH3D | | | | | | | | | | | | iBims-1 | | | | | | | | | | | | | |
| DINOv2 | 0.064 | 0.948 | 0.137 | 0.790 | 0.152 | 0.780 | 0.127 | 0.837 | 0.137 | 0.789 | 0.054 | 0.963 | 0.112 | 0.856 | 0.071 | 0.947 | 0.104 | 0.874 | 0.112 | 0.854 | | | | | |
| Ours | 0.053 | 0.963 | 0.091 | 0.902 | 0.075 | 0.939 | 0.088 | 0.923 | 0.092 | 0.900 | 0.044 | 0.973 | 0.091 | 0.899 | 0.060 | 0.962 | 0.085 | 0.924 | 0.091 | 0.899 | | | | | |
| GSO | | | | | | | | | | | | Sintel | | | | | | | | | | | | | |
| DINOv2 | 0.024 | 0.999 | 0.028 | 0.998 | 0.024 | 0.999 | 0.028 | 0.998 | 0.028 | 0.997 | 0.240 | 0.664 | 0.318 | 0.554 | 0.425 | 0.529 | 0.310 | 0.559 | 0.318 | 0.554 | | | | | |
| Ours | 0.018 | 1.000 | 0.026 | 1.000 | 0.019 | 1.000 | 0.026 | 1.000 | 0.026 | 1.000 | 0.204 | 0.715 | 0.305 | 0.580 | 0.342 | 0.604 | 0.298 | 0.573 | 0.305 | 0.580 | | | | | |
| DDAD | | | | | | | | | | | | DIODE | | | | | | | | | | | | | |
| DINOv2 | 0.147 | 0.789 | 0.230 | 0.598 | 0.224 | 0.703 | 0.215 | 0.647 | 0.231 | 0.595 | 0.071 | 0.927 | 0.179 | 0.731 | 0.128 | 0.849 | 0.159 | 0.774 | 0.179 | 0.730 | | | | | |
| Ours | 0.135 | 0.810 | 0.230 | 0.594 | 0.212 | 0.742 | 0.219 | 0.626 | 0.231 | 0.589 | 0.061 | 0.942 | 0.169 | 0.751 | 0.105 | 0.893 | 0.159 | 0.762 | 0.169 | 0.751 | | | | | |
| Spring | | | | | | | | | | | | HAMMER | | | | | | | | | | | | | |
| DINOv2 | 0.277 | 0.606 | 0.380 | 0.434 | 0.472 | 0.415 | 0.364 | 0.457 | 0.380 | 0.434 | 0.061 | 0.954 | 0.091 | 0.915 | 0.068 | 0.954 | 0.089 | 0.924 | 0.091 | 0.914 | | | | | |
| Ours | 0.245 | 0.638 | 0.385 | 0.436 | 0.431 | 0.495 | 0.373 | 0.430 | 0.385 | 0.435 | 0.053 | 0.962 | 0.080 | 0.933 | 0.060 | 0.967 | 0.079 | 0.941 | 0.080 | 0.932 | | | | | |

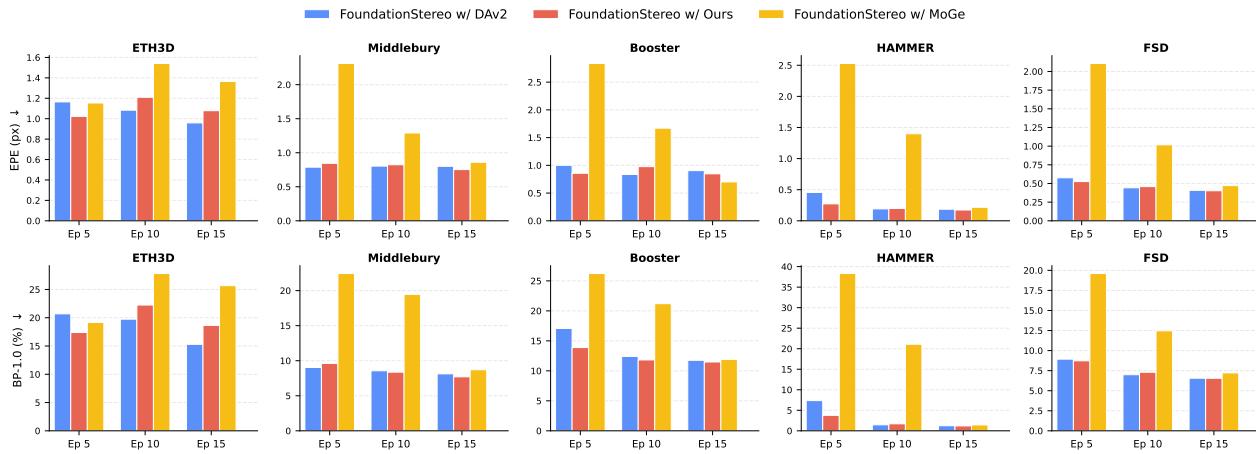


Figure 10. Epoch-wise comparison of FoundationStereo trained with different depth priors. We report EPE (top) and BP-1.0 (bottom) at epochs 5, 10, and 15 across five benchmarks.

Results. As shown in Tab. 3, models initialized with our encoder achieve consistent and significant improvements across all datasets compared to those based on DINOv2, demonstrating stronger generalization and enhanced intrinsic spatial understanding. By learning robust spatial representations through masked depth modeling, the encoder enables accurate depth reasoning even without depth inputs at inference time. These results confirm that our pretraining paradigm effectively distills 3D geometric knowledge into the encoder, improving its ability to infer depth structure from monocular images.

4.3 FoundationStereo with MDM Pretraining

We also use our MDM-pretrained LingBot-Depth model as a strong monocular depth prior in FoundationStereo [33]. Following the meta architecture, we adapt the RGB-only branch of LingBot-Depth model as the foundational depth priors and keep the rest of the architectures the same as their official implementation. Because our model architecture in MDM is similar to MoGe [29], we also use MoGe’s official implementation as an alternative to train a FoundationStereo with DepthAnythingV2 for comparison. Besides, we also train the vanilla FoundationStereo to obtain epoch-wise comparisons. The FSD dataset [33] is used for training. We keep all training hyperparameters the same for these three runs to make comparisons. To save the compute resource of training, we do not do iterative self data curation as what it was done in their original implementation. The total epochs of each run is 15.

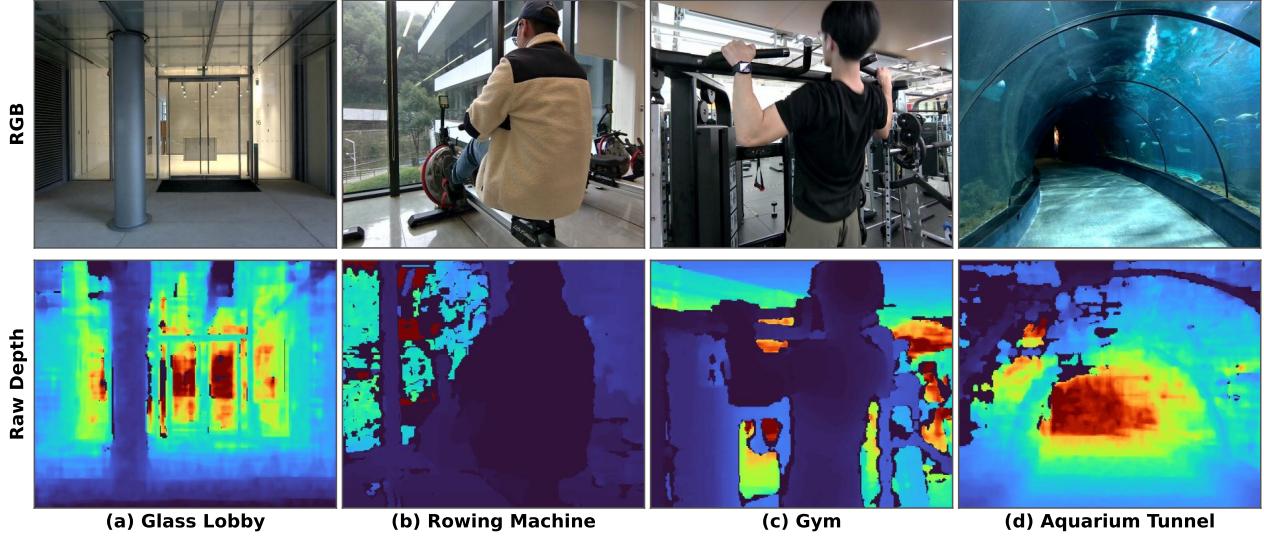


Figure 11. Video depth completion results under challenging scenarios. The raw depth maps from the Orbbee camera exhibit significant missing regions (shown in black) around transparent and reflective surfaces such as glass walls, windows, mirrors, and aquarium tunnels.

Evaluation Protocol. We follow the evaluation protocol of the three trained models on ETH3D [24], Middlebury [23], Booster [20], Hammer [11] and FoundationStereo dataset [33] and report the EPE (end-point-error) and BP-1.0 that computes the percentage of pixels where the disparity error is larger than 1.0 pixel.

Results. We report the performance of each model at 5-th, 10-th and 15-th epochs in Fig. 10. Several key observations emerge: (1) **Faster convergence.** At epoch 5, FoundationStereo with our pretrained encoder already achieves strong performance, outperforming the vanilla baseline on most benchmarks (*e.g.*, HAMMER EPE: 0.27 vs. 0.46, Booster EPE: 0.86 vs. 1.00). (2) **Training stability.** The MoGe-based variant exhibits significant instability in early training, with dramatically higher errors at epoch 5 (*e.g.*, HAMMER EPE: 2.53, Booster EPE: 2.84) that persist through epoch 10. (3) **Final performance.** At epoch 15, our variant achieves the best or comparable results across all benchmarks (Middlebury EPE: 0.75, HAMMER EPE: 0.17, FSD EPE: 0.40), confirming that MDM pretraining provides a more effective initialization for downstream stereo matching.

5 Extensions and Applications

In this section, we mainly showcase the extensions and applications built atop of our LingBot-Depth with a Orbbee Gemini-335 camera as the RGB-D input. There is no post-training in any of the experiments in this section. We hope our preliminary extensions could answer a question of why are RGB-D cameras important and how will RGB-D cameras shape the future in spatial perception.

5.1 Video Depth Completion

Even though our MDM pretraining was conducted on image data instead of the video inputs, we found that our LingBot-Depth model could significantly reduce the spatial-temporal inconsistency of the RGB-D cameras in several challenging scenarios. Figure 11 shows four challenging scenarios (*i.e.*, glass lobby, rowing machine, gym and aquarium tunnels) that are captured by ourselves. These videos are captured at 30 FPS in 640×480 resolution.

We make comparisons to the ZED-mini camera that is co-mounted in our portable capture system. As shown in Fig. 12, we compare our LingBot-Depth predictions against the ZED stereo camera depth, which serves as a higher-quality reference. Our model significantly improves upon the raw Orbbee depth by: (1) *filling large missing regions* caused by transparent and reflective surfaces where both structured-light (Orbbee) and stereo (ZED) sensors fail, such as glass walls in the lobby, windows near the rowing machine, mirrors in the gym, and the aquarium tunnel; (2)

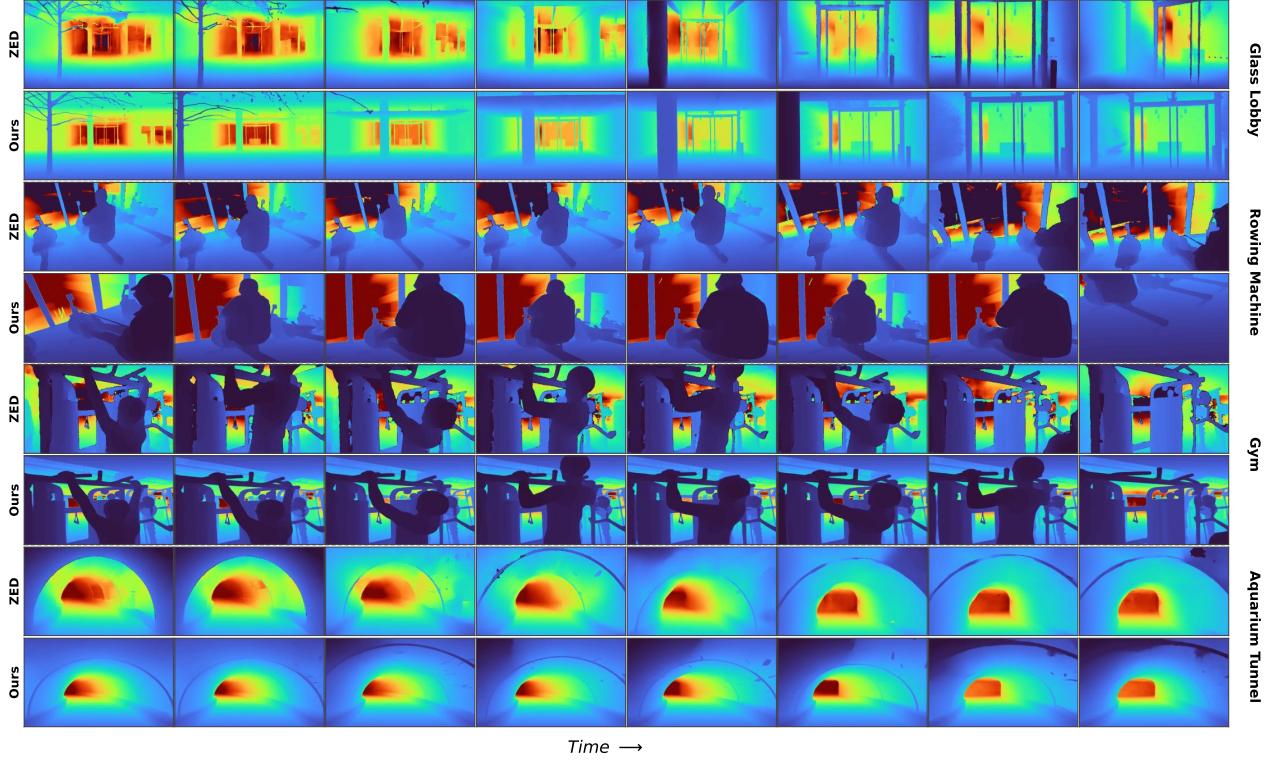


Figure 12. Video depth completion results compared with ZED stereo depth. For each scenario, the top row shows ZED depth and the bottom row shows our predictions. The ZED sensor suffers from significant missing regions on transparent and reflective surfaces, while our model produces spatially complete and temporally consistent depth maps throughout all sequences.

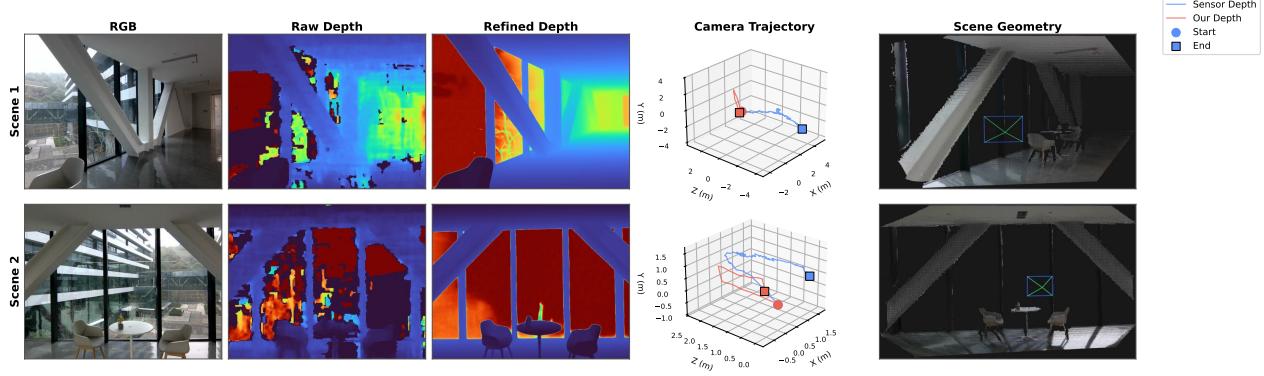
recovering fine-grained structures including thin objects (e.g., exercise equipment bars, ceiling pipes) and sharp object boundaries that are often noisy or absent in raw sensor outputs; and (3) maintaining temporal consistency across frames without explicit temporal modeling or video-specific training, producing smooth and stable depth sequences despite the per-frame inference. Notably, in the aquarium tunnel scenario, the ZED stereo camera almost entirely fails due to the refractive glass surfaces, while our model produces geometrically plausible depth throughout the sequence.

5.2 Online 3D Point Tracking

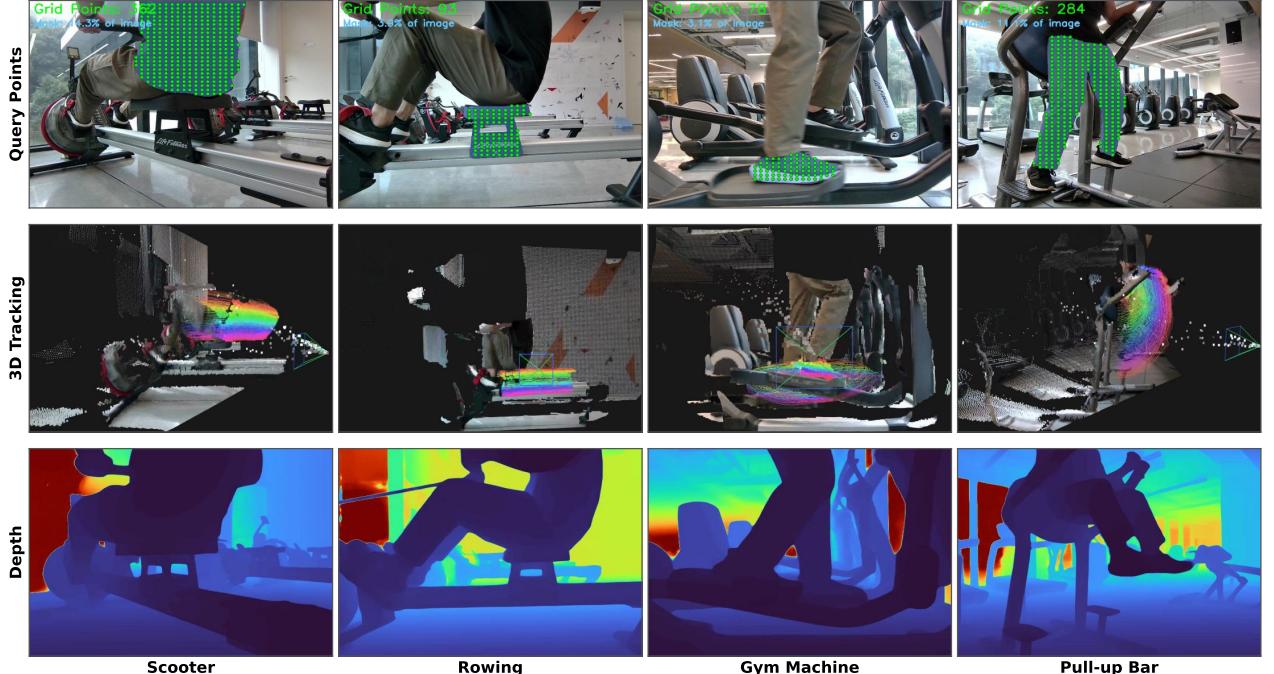
With the excellent performance on video depth completion, we leverage an off-the-shelf SpatialTrackerV2 [34] to obtain the camera poses and optionally track 3D points on dynamic objects. In our implementation, we use the online version of SpatialTrackerV2 [34] that does not depend on the VGGT [27]-Frontend for initial pose and depth estimation as the base, and tailored it as an RGB-D tracking baseline. Different from the SpatialTrackerV2’s original implementation that assumed the RGB-D inputs are with known camera extrinsics, we initialize the framewise extrinsics using SE-3 identity and mainly resort to the Bundle Adjustment (BA) on the tracked VO points to track the camera motion. There is no global BA applied to enable the efficiency. Besides, we did not finetune the SpatialTrackerV2.

Camera Tracking. As shown in Fig. 13a, we evaluate camera tracking on two indoor scenes with extensive glass surfaces where the raw depth sensor fails severely. Using the refined depth from our model, the SpatialTrackerV2 produces significantly smoother and more accurate camera trajectories compared to using raw sensor depth, which suffers from severe drift due to the missing depth regions.

Object Motion Tracking. We further demonstrate dynamic 3D point tracking on four scenarios with moving objects, as shown in Fig. 13b. Given a set of query points on the object of interest, SpatialTrackerV2 tracks their 3D trajectories using our refined depth. The rainbow-colored trails in the 3D visualizations reveal coherent motion patterns for each object, confirming that our depth predictions maintain sufficient geometric accuracy for downstream dynamic tracking tasks.



(a). Camera tracking and scene reconstruction. From left to right: RGB input, raw sensor depth, our refined depth, estimated camera trajectories (sensor depth in blue vs. our depth in red), and the reconstructed scene geometry.



(b). Dynamic 3D point tracking. Top: query points on the target object. Middle: 3D tracked trajectories (rainbow-colored by time). Bottom: corresponding depth maps.

Figure 13. Online 3D point tracking results using SpatialTrackerV2 with our refined depth.

5.3 Grasp Pose Generation in Real-World Robotics

We apply our LingBot-Depth to a real-world dexterous grasping pipeline, where accurate depth is critical for generating precise grasp poses.

Setup. Our system consists of a Rokae XMate-SR5 robotic arm equipped with an X Hand-1 dexterous hand, and an Orbtec Gemini 335 RGB-D camera for perception. Given an RGB-D observation, we first convert the depth into a point cloud and then predict an $N \times 22$ dexterous hand pose using a diffusion policy. The policy is conditioned on RGB features extracted by DINOv2 (ViT-L/14) and point cloud features from a Point Transformer, following a DP3-like architecture [38]. The model is trained on the HOI4D dataset [14], where human hand-object interactions are retargeted to the dexterous hand configuration via 3D keypoint correspondence.

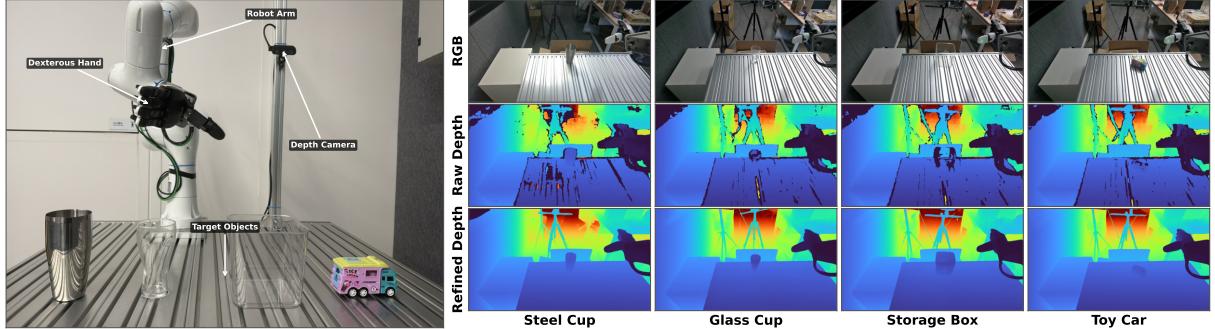


Figure 14. Qualitative results of the grasping experiment. Left: hardware setup with the robotic arm, dexterous gripper, and depth camera. Right: RGB, raw sensor depth, and our refined depth for four target objects. The raw depth is severely corrupted for reflective (steel cup) and transparent (glass cup, storage box) objects, while our method produces complete, geometrically accurate depth maps.

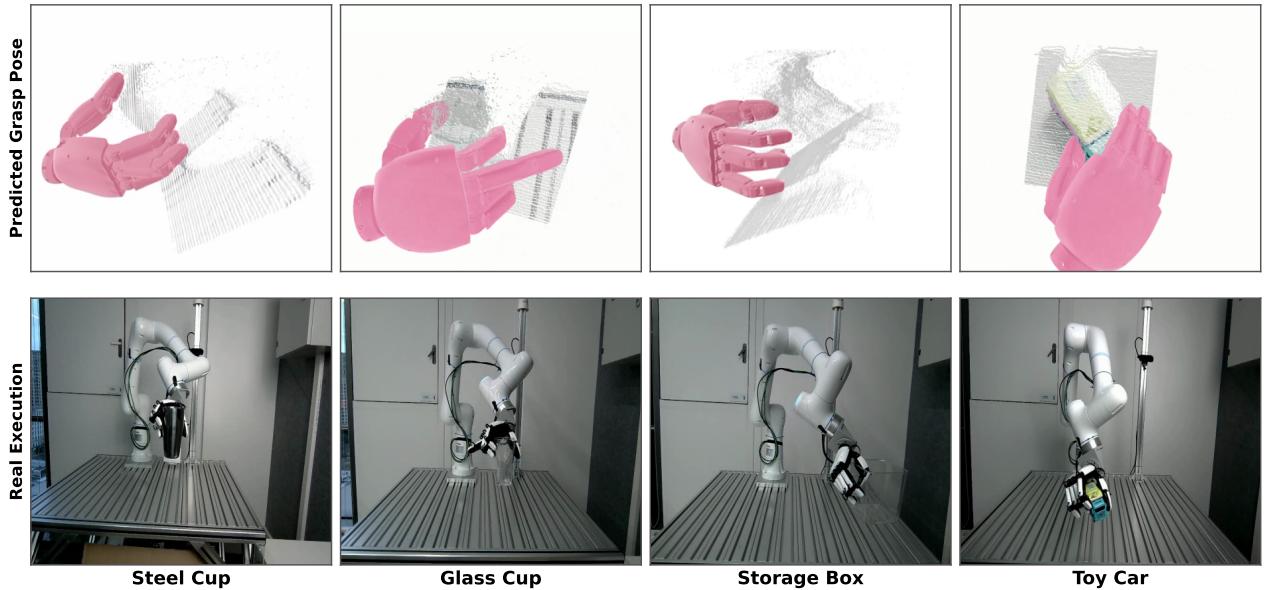


Figure 15. Grasp pose generation and real-world execution. Top: predicted grasp poses rendered as a dexterous hand overlaid on the point cloud reconstructed from our refined depth. Bottom: successful grasps executed by the robotic system on each target object.

Table 4. Grasping success rates (out of 20 trials) using our refined depth vs. raw sensor depth on four challenging objects. The transparent storage box is entirely ungraspable with raw depth due to severe depth corruption.

| | Total Times | Grasp with LingBot-Depth | Grasp with raw depth |
|-------------------------|-------------|--------------------------|----------------------|
| Stainless steel cup | 20 | 17 | 13 |
| Transparent cup | 20 | 16 | 12 |
| Toy Car | 20 | 16 | 9 |
| Transparent storage box | 20 | 10 | N/A |

Results. We evaluate grasping performance on four challenging objects, including transparent and reflective items for which raw depth sensors typically fail. Figure 15 shows the predicted grasp poses and their interaction with the point clouds reconstructed from our completed depth maps. As reported in Tab. 4, LingBot-Depth consistently improves grasping success rates compared to using raw sensor depth. Notably, the transparent storage box is entirely ungraspable with raw depth (N/A) due to severe depth corruption, whereas our model achieves a 50% success rate by producing geometrically plausible depth estimates, despite occasional inaccuracies on highly transparent surfaces. These results demonstrate that improved depth completion directly translates into more reliable robotic manipulation in real-world scenarios.

6 Conclusion

In this technical report, we investigate the problem of missing depth measurements in RGB-D cameras. Rather than treating these missing values purely as sensor failures, we leverage them as natural masks that reflect appearance ambiguities inherent in depth imaging systems, and propose an effective solution termed Masked Depth Modeling (MDM). Using 3 million self-curated RGB-D samples together with open-source depth datasets, we pretrain a large Vision Transformer to learn joint representations of pixel appearance and depth measurements. The resulting model, LingBot-Depth, demonstrates strong performance on both depth completion and monocular depth estimation, and further serves as a powerful monocular depth prior for FoundationStereo. Beyond static image tasks, we show that LingBot-Depth model produces high-quality video depth streams with strong spatial and temporal consistency. We further validate its effectiveness in downstream applications, including 3D point tracking and dexterous robotic grasping.

Acknowledgments

We gratefully acknowledge the optical testing team at Orbbee Inc. for their professional support.

References

- [1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15619–15629, 2023.
- [2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Eur. Conf. Comput. Vis.*, volume 7577, pages 611–625, 2012.
- [3] Qiyu Dai, Jiayao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *Eur. Conf. Comput. Vis.*, 2022.
- [4] Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *Adv. Neural Inform. Process. Syst.*, 2021.
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021.
- [6] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas Barlow McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022.
- [7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3354–3361, 2012.
- [8] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2482–2491, 2020.
- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15979–15988, 2022.
- [10] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):328–341, 2008.
- [11] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, Ales Leonardis, and Benjamin Busam. Is my depth ground-truth good enough? HAMMER – highly accurate multi-modal dataset for dense 3d scene regression. *CoRR*, abs/2205.04565, 2022.
- [12] Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the IBims-1 dataset. *Comput. Vis. Image Underst.*, 191:102877, 2020.

- [13] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17070–17080, 2025.
- [14] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4d egocentric dataset for category-level human-object interaction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21013–21022, 2022.
- [15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019.
- [16] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4981–4991, 2023.
- [17] Chuong V. Nguyen, Shahram Izadi, and David R. Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 524–530, 2012.
- [18] Maxime Oquab, Timothée Darzet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.
- [19] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar M. Parkhi, Richard A. Newcombe, and Carl Yuheng Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Int. Conf. Comput. Vis.*, pages 20076–20086, 2023.
- [20] Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Open challenges in deep stereo: The booster dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21168–21178, 2022.
- [21] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Ángel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Int. Conf. Comput. Vis.*, pages 10892–10902, 2021.
- [22] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Cleargrasp: 3d shape estimation of transparent objects for manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642, 2020.
- [23] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition: 36th German Conference, GCPR 2014*, pages 31–42, 2014.
- [24] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2538–2547, 2017.
- [25] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Eur. Conf. Comput. Vis.*, volume 7576, pages 746–760, 2012.
- [26] Igor Vasiljevic, Nicholas I. Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A dense indoor and outdoor depth dataset. *CoRR*, abs/1908.00463, 2019.
- [27] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5294–5306, 2025.
- [28] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5261–5271, 2025.
- [29] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. In *Adv. Neural Inform. Process. Syst.*, 2025.
- [30] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian A. Scherer. Tartanair: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916, 2020.

- [31] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025.
- [32] Songlin Wei, Haoran Geng, Jiayi Chen, Congyue Deng, Wenbo Cui, Chengyang Zhao, Xiaomeng Fang, Leonidas J. Guibas, and He Wang. D³roma: Disparity diffusion-based depth sensing for material-agnostic robotic manipulation. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 4944–4966, 2024.
- [33] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5249–5260, 2025.
- [34] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Int. Conf. Comput. Vis.*, 2025.
- [35] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Adv. Neural Inform. Process. Syst.*, 2024.
- [36] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Int. Conf. Comput. Vis.*, pages 12–22, 2023.
- [37] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3712–3722, 2018.
- [38] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Robotics: Science and Systems (RSS)*, 2024.
- [39] Yiming Zuo, Willow Yang, Zeyu Ma, and Jia Deng. OMNI-DC: Highly robust depth completion with multiresolution depth integration. In *Int. Conf. Comput. Vis.*, 2025.