Focusable Monocular Depth Estimation

Yuxin Du[1,2,*], **Tao Lin**[1,*], **Zile Zhong**[1], **Runting Li**[1], **Xiyao Chen**[1], **Jiting Liu**[1], **Chenglin Liu**[2], **Ying-Cong Chen**[2], **Yuqian Fu**[3], **Bo Zhao**[1,+]

[1]School of AI, Shanghai Jiao Tong University [2]HKUST (GZ) [3]KAUST
[*]These authors contributed equally to this work.
[+]Corresponding author.
Email: {yuxindu444,bo.zhao}@sjtu.edu.cn

Abstract

Monocular depth foundation models have substantially advanced full-image relative-depth estimation. Object-centric perception, however, places greater emphasis on depth accuracy for a specified target, particularly in its foreground and boundary regions. Most monocular depth estimators are trained and evaluated with image-wide objectives and metrics that do not distinguish the target relevant to a given perception task. We propose **FocusDepth**, a target-conditioned framework for **Focusable Monocular Depth Estimation (FDE)**, where a box or category-name prompt specifies the target whose depth prediction is refined within a full-image relative depth map. FocusDepth integrates geometric representations from Depth Anything variants with prompt-conditioned Segment Anything Model 3 (SAM3) features through **Multi-Scale Spatial-Aligned Fusion (MSSA)**, which maintains token-level spatial correspondence and performs independently parameterized routed and gated fusion at each depth feature scale. We further establish **FDE-Bench**, an image–target–depth benchmark constructed from five indoor and object-centric RGB-D datasets, comprising 252.9K training and 72.5K validation triplets spanning 972 target categories. On FDE-Bench, FocusDepth consistently improves depth accuracy for the specified target under both box and category-name prompting, with the largest gains in its foreground and boundary regions while retaining full-image relative-depth accuracy. On RLBench under box prompting, FocusDepth reduces AbsRel on the prompted object's boundary and foreground from 0.073 and 0.095 for DA3-ft to 0.049 and 0.056, respectively. Prompting the evaluated object rather than another object in the same image reduces median foreground AbsRel by 2.8–8.6% across DA2- and DA3-based variants, providing direct evidence that FocusDepth conditions its refinement on the specified target.

1 Introduction

Monocular depth estimation [24, 3, 1] aims to recover dense scene geometry from a single RGB image and serves as a fundamental visual capability in many vision systems, including robotics, embodied AI, and augmented reality. The field has been extensively studied under supervised and self-supervised learning paradigms [44, 10–12], with later work improving robustness through cross-dataset training, stronger architectures, and generative priors [28, 45, 43, 17, 13, 42]. More recently, depth foundation models such as Depth Anything [36, 37, 19], UniDepth [26, 27], and Metric3D [38, 14] have substantially advanced cross-domain generalization by leveraging large-scale and diverse training data. Despite this progress, monocular depth estimation remains predominantly image-centric: models are typically trained and evaluated by aggregating errors over all pixels in the image. This uniform, image-wide treatment contrasts with object-centric perception, which places greater emphasis on

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

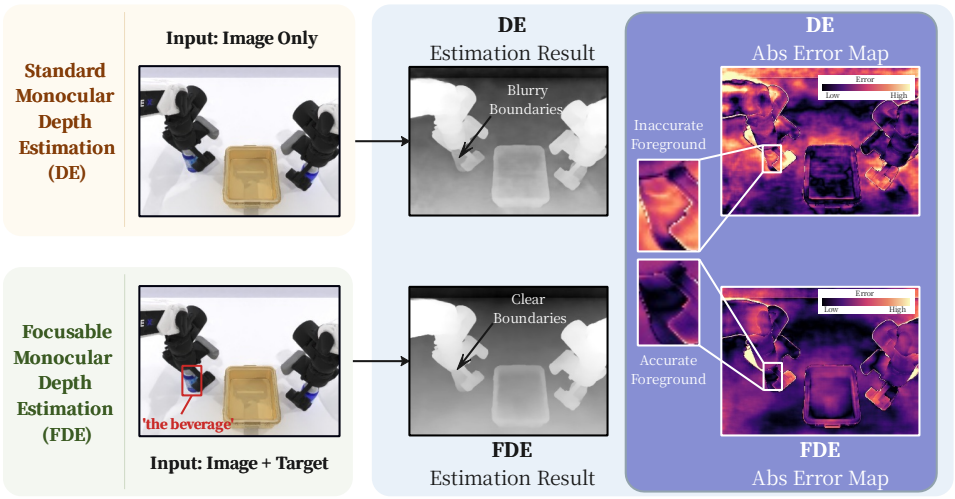


Figure 1: Illustration of **FocusDepth** for Focusable Monocular Depth Estimation (FDE). Standard monocular depth estimation produces a full-image relative depth map without specifying a target. Given the same image and a target prompt, FocusDepth directs refinement toward the prompted object's foreground and boundary regions while retaining full-image relative-depth accuracy.

depth accuracy for a specified target, particularly in its foreground and boundary regions. This mismatch is also relevant to target-centric tasks such as robot manipulation [4, 15, 21, 22, 32, 31] and spatial visual question answering [7, 8, 5], where reasoning about a specified object draws on accurate depth at its boundaries and its relative depth with respect to surrounding objects. Prior task-aware depth estimation has shown that uniform image-wide objectives can underweight foreground objects in downstream 3D perception [33]. Consequently, strong full-image depth performance does not ensure comparable accuracy for a task-relevant target. Because the relevant target can change even when the image remains fixed, target-centric depth estimation needs an explicit interface for directing refinement toward the current target.

To address this formulation gap, we introduce **Focusable Monocular Depth Estimation (FDE)**, a target-conditioned setting in which a box or category-name prompt specifies the target to be refined within a full-image relative depth map. As illustrated in Figure 1, FDE prioritizes depth accuracy on the target foreground and boundary while retaining full-image relative-depth accuracy.

To study FDE, we propose **FocusDepth**, a target-conditioned framework designed to refine depth prediction for a specified target within a full-image relative depth map. Our key intuition is to combine the complementary priors of promptable segmentation and monocular depth foundation models: Segment Anything Model 3 (SAM3) [6] provides prompt-grounded spatial selectivity for identifying user-specified target regions, while the Depth Anything family (DAs), such as DA2 [37] and DA3 [19], provides a strong pretrained prior over dense scene geometry. However, directly fusing these models is nontrivial, as SAM3 is optimized for 2D prompt-driven localization whereas DAs are optimized for dense full-image depth representations. FocusDepth injects prompt-conditioned target cues into Depth Anything representations through controlled, spatially aligned fusion. We introduce **MSSA** (**M**ulti-**S**cale **S**patial-**A**ligned Fusion), which injects prompt-conditioned cues into DAs geometry tokens through spatially aligned token-level fusion, adapts them across depth feature scales, and uses gated conditional fusion to control prompt-conditioned corrections. This design encourages target-relevant regions, including foreground and boundary areas, to receive focused refinement while retaining the full-image depth representation.

To make FDE measurable, we further establish **FDE-Bench**, a benchmark suite and evaluation protocol that reformulates existing depth datasets into image–target–depth triplets. Unlike conventional depth benchmarks that primarily report full-image error, FDE-Bench evaluates depth accuracy on the specified foreground and target boundary, while reporting full-image metrics to assess the scene-wide effect of target-conditioned refinement. The benchmark covers diverse RGB-D sources, including indoor scenes, tabletop object data, and simulated embodied-scene observations from NYU v2 [29], TUM RGB-D [30], YCB-Video [35], RLBench [16], and RoboTwin [25], containing

252.9K/72.5K train/val triplets. This protocol enables systematic comparison of standard monocular depth models and prompt-conditioned methods under target-centric criteria, assessing foreground accuracy, boundary fidelity, and full-image relative-depth accuracy.

Across box and category-name prompting, FocusDepth consistently improves depth accuracy for the specified target, with the largest gains in its foreground and boundary regions while retaining full-image relative-depth accuracy. In same-image target-switch experiments, prompting the evaluated object rather than another object reduces median foreground AbsRel by 2.8–8.6% across DA2- and DA3-based variants, demonstrating that the prompt determines which target receives refinement.

Our contributions are threefold:

- We propose **FocusDepth**, a target-conditioned framework that uses a box or category-name prompt to refine depth prediction for a specified target within a full-image relative depth map, with refinement focused on its foreground and boundary regions.
- We introduce **Multi-Scale Spatial-Aligned Fusion (MSSA)**, which integrates geometric representations from Depth Anything with prompt-conditioned SAM3 features while maintaining token-level spatial correspondence. Independently parameterized routed and gated fusion at each depth feature scale injects target cues into multi-scale depth representations and directs refinement toward the specified target.
- We establish **FDE-Bench**, an image–target–depth benchmark constructed from five indoor and object-centric RGB-D datasets. Experiments under both box and category-name prompting show that FocusDepth improves foreground and boundary depth accuracy for the specified target while retaining full-image relative-depth accuracy; same-image target switching further demonstrates that the prompt determines which object receives refinement.

2 Related Work

Monocular depth estimation and depth foundation models. Monocular depth estimation has evolved from supervised and self-supervised pipelines to more transferable architectures trained across diverse datasets [10–12, 28, 45, 43]. These developments have progressively improved depth accuracy, cross-domain robustness, and dense depth prediction, laying the groundwork for recent depth foundation models. Current leading models include UniDepth [26] and UniDepthV2 [27], Metric3D [38] and Metric3Dv2 [14], and the Depth Anything family [36, 37, 19]. By leveraging large-scale training data, metric supervision, synthetic data, or auto-annotated depth, these models achieve strong zero-shot generalization and robust whole-image depth prediction across diverse scenes. These models provide strong full-image depth priors. Their standard formulation remains image-conditioned, and evaluation aggregates error over the full image plane rather than testing refinement for a specified target.

Task-aware and conditioned depth estimation. Related methods introduce task, spatial, semantic, or geometric conditions into depth estimation. ForeSeE separates foreground and background prediction for task-aware depth estimation [33]. Mask-guided refinement uses externally supplied masks to improve object boundaries and hole regions [18], while Background Prompting improves object depth by placing segmented objects on learned backgrounds [2]. Language-guided methods use scene descriptions to provide metric or semantic depth priors [39, 40]; Iris further conditions diffusion-based depth estimation on text and reports larger improvements in described regions [41]. BriGeS combines depth and segmentation foundation models for full-image enhancement [23], while other methods introduce sparse depth or external geometry as prompts [20, 34]. FocusDepth uses a box or category-name prompt as an explicit target interface and performs target-conditioned refinement within a full-image relative depth map. Its evaluation measures target foreground and boundary accuracy and uses same-image target switching to test whether the prompt determines which object is refined.

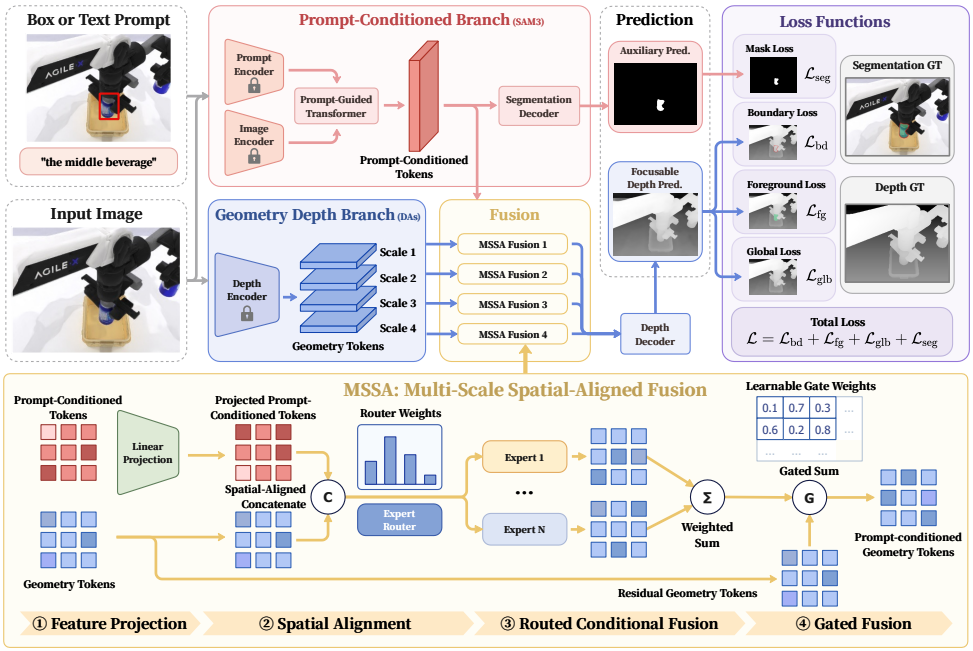


Figure 2: Overall framework of FocusDepth. The Depth Anything branch extracts multi-scale geometry tokens, while the SAM3 branch produces prompt-conditioned tokens for the specified target. MSSA maintains token-level spatial correspondence and integrates the target cues into multi-scale depth representations through expert routing and gated fusion, enabling target-conditioned refinement within a full-image relative depth map. Training uses foreground, boundary, and full-image depth losses together with auxiliary segmentation supervision.

3 Method

3.1 Problem Formulation

In FDE, a model takes an input image $I \in \mathbb{R}^{H\times W\times 3}$ and a prompt p specifying a target, and predicts a full-image relative depth map conditioned on that target. We consider two prompt types: a box prompt and a category-name prompt. The FDE objective assigns additional emphasis to the target's foreground and boundary regions while retaining supervision over the full image.

FocusDepth is formulated as $\hat{D}, \hat{M} = \mathcal{F}(I, p)$, where $\hat{D} \in \mathbb{R}^{H\times W}$ is the predicted full-image relative depth map and $\hat{M} \in [0,1]^{H\times W}$ is an auxiliary prompt-conditioned target mask. FocusDepth implements the FDE objective through target-conditioned feature fusion and region-aware supervision.

3.2 Overall Framework

Figure 2 illustrates the overall framework of FocusDepth. FocusDepth uses DA2 [37] and DA3 [19] as geometry backbones and SAM3 [6] as the prompt-conditioned branch for encoding the specified target. The geometry branch extracts multi-scale depth representations, while the prompt-conditioned branch produces target-conditioned tokens from the image and prompt. MSSA spatially aligns and fuses these tokens with the geometry tokens at each feature scale to perform target-conditioned depth refinement.

Given an input image I, the geometry depth branch extracts a hierarchy of encoder features $X^g = \{X^g_s\}_{s=1}^{S}, X^g_s \in \mathbb{R}^{N\times C_g}$, where s indexes the feature scale. We use $S = 4$ feature scales from the geometry encoder. The prompt-conditioned branch takes the image and prompt p as input and produces a prompt-conditioned token feature $X^p \in \mathbb{R}^{N\times C_p}$. The two branches share the same token count N because they process the same image over an identical patch grid of ViT [9]. FocusDepth applies MSSA at each geometry scale: $Y = \mathrm{MSSA}(X^g, X^p)$, where each scale has independent

fusion parameters. The fused features $Y = \{Y_s\}_{s=1}^{S}$ are passed to the depth decoder to predict $\hat{D}$. An auxiliary segmentation decoder on the prompt-conditioned branch predicts $\hat{M}$ during training phase.

3.3 Multi-Scale Spatial-Aligned Fusion

MSSA is the target-conditioned fusion module of FocusDepth. It injects prompt-conditioned cues into geometry tokens in a spatially adaptive manner while preserving the pretrained geometry prior. MSSA is applied independently at each geometry encoder scale.

Scale-wise spatial alignment. Under the shared setting of input image size 1008 and patch size 14 of ViT [9], the geometry depth branch and the prompt-conditioned branch produce image tokens on the same regular patch grid. Thus, their tokens are in one-to-one spatial correspondence. At scale s, the geometry depth branch provides geometry tokens $X_s^g \in \mathbb{R}^{N \times C_g}$, while the prompt-conditioned branch provides the prompt-conditioned tokens $X^p \in \mathbb{R}^{N \times C_p}$. We project X^p into the scale-specific geometry-token space: $\tilde{X}_s^p = X^p W_s^p$, where $W_s^p \in \mathbb{R}^{C_p \times C_g}$ is a learnable projection. The projected prompt-conditioned tokens and geometry tokens are then concatenated token-wise: $Z_s = [X_s^g \,\|\, \tilde{X}_s^p] \in \mathbb{R}^{N \times 2C_g}$. This spatial-aligned concatenation allows prompt-conditioned target cues to guide refinement at the corresponding spatial locations at each geometry scale.

Routed conditional fusion. Different image locations can require different prompt-conditioned corrections. To provide conditional fusion capacity, MSSA applies a lightweight MoE layer to the fused tokens Z_s: $F_s = \mathrm{MoE}_s(Z_s) \in \mathbb{R}^{N \times C_g}$. Each scale uses an independently parameterized MoE with $E = 4$ experts. Its router computes input-dependent mixtures of expert outputs, while the experts learn fusion patterns without predefined regional roles.

Gated fusion output. A learnable gate balances the routed prompt-conditioned feature and the original geometry representation: $G_s = \sigma(F_s W_s^g)$, where $W_s^g \in \mathbb{R}^{C_g \times 1}$ and $\sigma(\cdot)$ denotes the Sigmoid function. The output feature is $Y_s = G_s \odot F_s + (1 - G_s) \odot X_s^g$. This gated formulation injects prompt-conditioned corrections selectively while retaining the pretrained geometry prior.

3.4 Learning Objective

We use the target foreground, target boundary, and full-image regions defined by FDE-Bench. For each training triplet, we follow the notation in Appendix Section A.1.3: D is the ground-truth depth, M is the foreground mask, V is the valid-depth mask, and V_{fg}, V_{bd}, and V_{glb} denote the valid foreground, boundary, and global regions.

In the relative depth setting, we align the prediction to the ground truth by solving a per-image scale a and shift b over valid pixels: $\tilde{D} = a\hat{D} + b$. Regional depth supervision is then computed as

$$\mathcal{L}_{\mathrm{fg}} = \mathcal{L}_{\mathrm{depth}}(\tilde{D}, D; V_{\mathrm{fg}}), \qquad \mathcal{L}_{\mathrm{bd}} = \mathcal{L}_{\mathrm{depth}}(\tilde{D}, D; V_{\mathrm{bd}}), \qquad \mathcal{L}_{\mathrm{glb}} = \mathcal{L}_{\mathrm{depth}}(\tilde{D}, D; V_{\mathrm{glb}}), \quad (1)$$

where $\mathcal{L}_{\mathrm{depth}}$ combines masked MSE and gradient losses within the specified region. For DA2 [37], we apply a disparity-space processing step before computing the loss. The auxiliary mask prediction is supervised by $\mathcal{L}_{\mathrm{seg}} = \mathcal{L}_{\mathrm{BCE}}(\hat{M}, M) + \mathcal{L}_{\mathrm{Dice}}(\hat{M}, M)$. The final objective is $\mathcal{L} = \mathcal{L}_{\mathrm{glb}} + \mathcal{L}_{\mathrm{fg}} + \mathcal{L}_{\mathrm{bd}} + \mathcal{L}_{\mathrm{seg}}$. We assign unit weight to the three regional depth losses and the auxiliary segmentation loss. Each regional depth loss is normalized over its valid pixels.

3.5 Two-Stage Training Strategy

To stabilize adaptation between the pretrained geometry and prompt-conditioned components, we train FocusDepth in two stages. In Stage 1, only MSSA is optimized, while all pretrained modules are frozen. This lets MSSA learn to align geometry tokens with prompt-conditioned tokens without disturbing the pretrained backbones. In Stage 2, we keep the depth encoder, the SAM3 image encoder, and the SAM3 prompt encoder frozen, and optimize MSSA, the depth decoder, the SAM3 prompt-guided transformer, and the auxiliary segmentation decoder. This separates cross-model alignment from target-conditioned depth adaptation and improves training stability.

4 Proposed FDE-Bench

4.1 Benchmark Construction

FDE-Bench organizes target-centric depth evaluation around the *image–target–depth triplet*, comprising an RGB image, one specified target region, and the corresponding ground-truth depth map. A source image may yield multiple triplets with different targets. We construct FDE-Bench from five indoor and object-centric RGB-D datasets: NYU v2 [29], RLBench [16], YCB-Video [35], TUM RGB-D [30], and RoboTwin [25].

Figure 3 summarizes the three-stage construction pipeline. **Region selection.** We use source-provided instance or semantic masks when available; otherwise, SAM automatic mask generation provides pseudo masks. Both types of masks are filtered using the same area protocol, and pseudo-mask entries are marked in the metadata. **Prompt generation.** Each retained mask defines the target foreground, its tight bounding box provides the box prompt, and its semantic class name, when available, provides the category-name prompt. **Entry assembly.** We retain the original RGB image and depth map and attach one target specification to each triplet, storing the target mask for region-aware evaluation. Triplets from the same source image therefore share the RGB-D observation while specifying different targets.

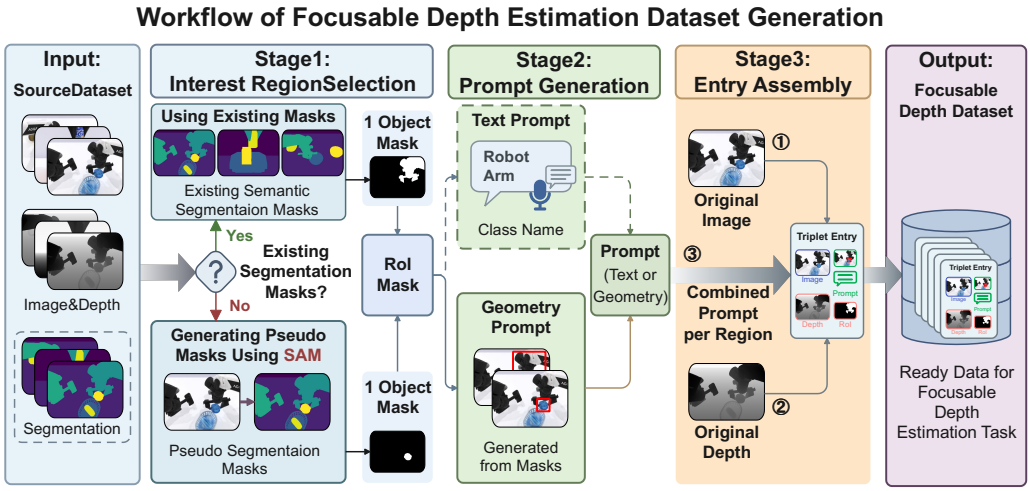


Figure 3: Construction pipeline of FDE-Bench. Existing depth datasets are converted from heterogeneous annotations into standardized image–target–depth triplets for unified region-aware evaluation.

FDE-Bench contains 20.5K training and 8.2K validation images, yielding 252.9K training and 72.5K validation triplets and covering 972 target categories with semantic labels (Appendix Table 6). Box prompting is available across all five datasets, while category-name prompting is available on NYU v2, RLBench, and YCB-Video. All entries derived from the same image or sequence are assigned to the same split.

4.2 Region-aware Evaluation Metrics

For each triplet, we report **AbsRel** and δ_1 over valid-depth pixels in three regions: the target foreground, the target boundary, and the full image. The boundary region is a band around the target mask with radius $r = 10$ pixels, induced by morphological dilation and erosion. Metrics are computed in a per-target manner. For relative-depth evaluation, a single scale-shift alignment is estimated over all valid pixels in the full image and then applied unchanged when computing the foreground, boundary, and full-image metrics. Dataset-specific statistics, split protocols, and evaluation details are provided in Appendix A.1.

Table 1: Box-prompt results on RLBench, RoboTwin, and NYU v2. We compare zero-shot depth foundation models, fine-tuned DA2/DA3 baselines, and FocusDepth variants across boundary, foreground, and global regions.

Dataset	Method	Boundary region		Foreground region		Global region	
		$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$
RLBench [16]	UniDepth-v1 [26]	0.597 (0.386, 0.800)	0.247	0.542 (0.008, 0.997)	0.234	0.665 (0.574, 0.820)	0.214
	UniDepth-v2 [27]	**0.871** (**0.572**, **0.996**)	**0.123**	**0.997** (**0.355**, **1.000**)	**0.127**	**0.921** (**0.686**, **0.958**)	**0.097**
	Metric3D-v2 [14]	0.707 (0.470, 0.894)	0.185	0.727 (0.081, **1.000**)	0.198	0.815 (0.644, 0.903)	0.165
	DA2 [37]	0.922 (0.680, 0.996)	0.106	0.996 (0.694, 1.000)	0.097	0.917 (0.726, 0.957)	0.106
	DA2-ft	0.982 (0.889, 1.000)	0.069	1.000 (0.949, 1.000)	0.066	0.964 (0.953, 0.990)	0.054
	FocusDepth(DA2)	**0.996** (**0.960**, 1.000)	**0.054**	1.000 (**0.991**, 1.000)	**0.050**	**0.992** (**0.979**, **0.998**)	**0.045**
	DA3 [19]	0.790 (0.488, 0.985)	0.148	0.944 (0.103, 1.000)	0.147	0.889 (0.648, 0.954)	0.134
	DA3-ft	0.975 (0.804, 1.000)	0.073	1.000 (0.709, 1.000)	0.095	0.983 (0.921, 0.997)	0.042
	FocusDepth(DA3)	**0.996** (**0.949**, 1.000)	**0.049**	1.000 (**0.977**, 1.000)	**0.056**	**0.996** (**0.974**, **0.999**)	**0.030**
RoboTwin [25]	UniDepth-v1 [26]	0.751 (0.502, 0.929)	0.190	0.840 (0.287, 0.995)	0.151	0.930 (0.812, 0.975)	0.102
	UniDepth-v2 [27]	**0.919** (0.689, **0.987**)	**0.100**	**0.997** (**0.804**, **1.000**)	**0.073**	**0.983** (0.957, **0.993**)	0.056
	Metric3D-v2 [14]	0.905 (**0.703**, 0.979)	0.101	0.988 (0.653, **1.000**)	0.081	**0.983** (**0.959**, 0.992)	**0.051**
	DA2 [37]	0.954 (0.571, 0.991)	0.095	0.993 (0.604, 1.000)	0.082	0.984 (0.849, 0.997)	0.062
	DA2-ft	0.957 (0.680, 0.995)	0.086	0.987 (0.679, 1.000)	0.084	0.989 (0.864, 0.997)	0.053
	FocusDepth(DA2)	**0.976** (**0.735**, **0.998**)	**0.070**	**0.995** (**0.878**, 1.000)	**0.065**	**0.994** (**0.880**, **0.999**)	**0.047**
	DA3 [19]	0.889 (0.565, 0.987)	0.119	0.973 (0.339, 1.000)	0.115	0.962 (0.842, 0.989)	0.077
	DA3-ft	0.949 (0.772, 0.995)	0.082	0.992 (0.793, 1.000)	0.075	0.985 (0.960, 0.997)	0.037
	FocusDepth(DA3)	**0.979** (**0.903**, **0.998**)	**0.058**	**0.998** (**0.955**, 1.000)	**0.055**	**0.995** (**0.977**, **0.999**)	**0.029**
NYU v2 [29]	UniDepth-v1 [26]	**1.000** (**0.960**, 1.000)	0.054	1.000 (**0.995**, 1.000)	0.045	0.973 (0.945, 0.987)	0.056
	UniDepth-v2 [27]	0.998 (0.948, 1.000)	**0.047**	1.000 (0.993, 1.000)	**0.037**	**0.977** (**0.949**, **0.990**)	**0.046**
	Metric3D-v2 [14]	0.996 (0.916, 1.000)	0.059	1.000 (0.979, 1.000)	0.051	0.974 (0.928, **0.990**)	0.059
	DA2 [37]	0.998 (0.946, 1.000)	0.058	1.000 (0.991, 1.000)	0.049	0.975 (**0.941**, 0.991)	**0.057**
	DA2-ft	**1.000** (0.971, 1.000)	0.058	1.000 (0.997, 1.000)	0.051	**0.979** (0.938, **0.995**)	0.061
	FocusDepth(DA2)	**1.000** (**0.976**, 1.000)	**0.055**	1.000 (**0.998**, 1.000)	**0.048**	**0.979** (**0.941**, **0.995**)	0.059
	DA3 [19]	0.970 (0.869, 1.000)	0.074	1.000 (0.986, 1.000)	0.055	0.927 (0.857, 0.956)	0.091
	DA3-ft	**1.000** (0.972, 1.000)	0.051	1.000 (0.997, 1.000)	0.046	0.977 (0.937, 0.993)	**0.053**
	FocusDepth(DA3)	**1.000** (**0.977**, 1.000)	**0.049**	1.000 (**0.998**, 1.000)	**0.044**	**0.978** (**0.939**, **0.995**)	**0.053**

5 Experiments

5.1 Experimental Setup

We evaluate FocusDepth(DA2) and FocusDepth(DA3) on FDE-Bench using the dataset scope, prompt settings, and region-aware evaluation metrics defined in Section 4. The framework design, region-aware learning objective, and training strategy follow Section 3. Unless otherwise specified, all experiments are conducted on 4 NVIDIA A100 80GB GPUs. We next describe the comparison settings and report quantitative and qualitative results. To establish a baseline for the FDE capability of current representative monocular depth foundation models, we evaluate UniDepth-v1, UniDepth-v2, Metric3D-v2, Depth Anything 2 (DA2), and Depth Anything 3 (DA3) in the zero-shot setting. We further report the fine-tuning results of DA2 and DA3, and compare them with FocusDepth under the same training resources and data, with detailed hyperparameter settings in Appendix Table 7 and Appendix Table 8.

5.2 Main Results

Table 1 and Table 2 report the main box-prompt and text-prompt comparison on NYU v2, RLBench, RoboTwin, and YCB-Video. Appendix Table 9 provides the remaining box-prompt results on TUM RGB-D and YCB-Video. Following the per-target evaluation protocol in FDE-Bench, we first compute metrics for each image–target–depth triplet and then report the median over target-level metrics, with Q25/Q75 shown for δ_1. Across all tables, higher δ_1 and lower AbsRel indicate better performance.

Zero-shot baselines. Table 1 shows that current monocular depth foundation models already provide strong whole-image geometry in several settings, but their target-centric behavior is not consistently aligned with FDE. On NYU v2, zero-shot baselines are close to saturation in δ_1, making differences mainly visible through AbsRel and lower-quartile statistics. On RLBench and

Table 2: Text-prompt results on RLBench, NYU v2, and YCB-Video. We compare zero-shot depth foundation models, fine-tuned DA2/DA3 baselines, and FocusDepth variants across boundary, foreground, and global regions.

Dataset	Method	Boundary region		Foreground region		Global region	
		$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$
RLBench [16]	UniDepth-v1 [26]	0.601 (0.423, 0.775)	0.247	0.577 (0.148, 0.958)	0.224	0.666 (0.575, 0.822)	0.212
	UniDepth-v2 [27]	**0.859** (**0.579**, **0.990**)	**0.121**	**0.991** (**0.454**, **1.000**)	**0.117**	**0.925** (**0.695**, **0.958**)	**0.093**
	Metric3D-v2 [14]	0.716 (0.486, 0.870)	0.183	0.753 (0.271, 0.998)	0.185	0.817 (0.648, 0.905)	0.163
	DA2 [37]	0.895 (0.691, 0.992)	0.110	0.989 (0.781, 1.000)	0.090	0.921 (0.728, 0.957)	0.103
	DA2-ft	0.976 (0.888, 0.999)	0.069	0.999 (0.949, 1.000)	0.061	0.964 (0.953, 0.991)	0.054
	FocusDepth(DA2)	**0.994** (**0.961**, **1.000**)	**0.053**	**1.000** (**0.988**, 1.000)	**0.046**	**0.993** (**0.980**, **0.998**)	**0.044**
	DA3 [19]	0.795 (0.507, 0.971)	0.145	0.910 (0.294, 1.000)	0.139	0.893 (0.655, 0.955)	0.129
	DA3-ft	0.973 (0.824, 0.999)	0.068	0.998 (0.772, 1.000)	0.088	0.984 (0.923, 0.997)	0.041
	FocusDepth(DA3)	**0.995** (**0.959**, **1.000**)	**0.045**	**1.000** (**0.977**, 1.000)	**0.050**	**0.996** (**0.978**, **0.999**)	**0.029**
NYU v2 [29]	UniDepth-v1 [26]	**0.998** (**0.952**, 1.000)	0.054	1.000 (**0.991**, 1.000)	0.046	0.973 (0.945, 0.987)	0.056
	UniDepth-v2 [27]	0.994 (0.943, 1.000)	**0.048**	1.000 (0.989, 1.000)	**0.038**	**0.977** (**0.949**, **0.990**)	**0.046**
	Metric3D-v2 [14]	0.990 (0.912, 1.000)	0.060	1.000 (0.970, 1.000)	0.052	0.974 (0.928, **0.990**)	0.059
	DA2 [37]	0.995 (0.939, 1.000)	0.059	1.000 (0.986, 1.000)	0.050	0.975 (**0.941**, 0.991)	**0.057**
	DA2-ft	**1.000** (0.960, 1.000)	0.059	1.000 (0.993, 1.000)	0.052	**0.979** (0.938, **0.995**)	0.061
	FocusDepth(DA2)	**1.000** (**0.963**, 1.000)	**0.057**	1.000 (**0.995**, 1.000)	**0.049**	**0.979** (0.940, **0.995**)	0.060
	DA3 [19]	0.959 (0.865, 1.000)	0.077	1.000 (0.979, 1.000)	0.057	0.927 (0.857, 0.956)	0.091
	DA3-ft	**1.000** (0.961, 1.000)	0.052	1.000 (0.992, 1.000)	0.047	0.977 (0.937, 0.993)	**0.053**
	FocusDepth(DA3)	**1.000** (**0.967**, 1.000)	**0.050**	1.000 (**0.995**, 1.000)	**0.044**	**0.978** (**0.938**, **0.995**)	0.054
YCB-Video [35]	UniDepth-v1 [26]	**1.000** (0.985, 1.000)	0.041	1.000 (0.997, 1.000)	0.037	0.993 (0.981, 0.997)	0.045
	UniDepth-v2 [27]	0.997 (0.966, 1.000)	**0.036**	1.000 (0.994, 1.000)	**0.026**	0.996 (**0.991**, 0.998)	**0.031**
	Metric3D-v2 [14]	**1.000** (**0.986**, 1.000)	0.041	1.000 (**0.998**, 1.000)	0.031	**0.997** (0.988, **0.999**)	0.042
	DA2 [37]	0.998 (0.963, 1.000)	0.045	1.000 (0.993, 1.000)	0.037	0.994 (0.972, 0.998)	0.040
	DA3 [19]	0.993 (0.938, 1.000)	0.053	0.999 (0.986, 1.000)	0.044	0.992 (0.961, 0.997)	0.047
	DA3-ft	**1.000** (0.988, 1.000)	0.042	**1.000** (**0.999**, 1.000)	0.035	0.997 (0.984, **0.999**)	0.044
	FocusDepth(DA3)	**1.000** (**0.990**, 1.000)	**0.034**	**1.000** (**0.999**, 1.000)	**0.024**	**0.998** (**0.993**, **0.999**)	**0.032**

RoboTwin, however, the gap between global performance and target-relevant boundary or foreground performance becomes more apparent. This suggests that strong image-centric depth estimation does not by itself ensure reliable focusable depth estimation on manipulation-oriented and object-centric scenes.

FocusDepth gains. Compared with the corresponding globally fine-tuned DA variants, FocusDepth more consistently improves the prompted target regions while retaining full-image relative-depth accuracy. The clearest gains appear on RLBench: under box prompts, FocusDepth(DA3) reduces boundary AbsRel from 0.073 to 0.049, foreground AbsRel from 0.095 to 0.056, and global AbsRel from 0.042 to 0.030 relative to DA3-ft. RoboTwin shows the same pattern, with FocusDepth(DA3) improving boundary, foreground, and global AbsRel over DA3-ft from 0.082/0.075/0.037 to 0.058/0.055/0.029. Under box prompting, FocusDepth(DA3) mainly improves the boundary and foreground regions over DA3-ft on NYU v2, while matching DA3-ft in global AbsRel at 0.053. FocusDepth(DA2) follows a similar trend over DA2-ft, indicating that the benefit is not tied to a single backbone. The category-name-prompt comparison in Table 2 shows consistent foreground and boundary AbsRel improvements over the corresponding fine-tuned DA baselines on RLBench, NYU v2, and YCB-Video. The appendix results further support this conclusion: the remaining box-prompt results in Table 9 show clear improvements on TUM RGB-D and YCB-Video. Overall, these results support the FocusDepth objective: prompt-conditioned refinement most reliably improves local target regions, with limited trade-offs in full-image relative-depth accuracy in most evaluated settings.

5.3 Same-Image Target Switching

We examine whether changing the prompt redirects depth refinement between objects in the same image. For each object pair (A, B) on RLBench, we generate two depth predictions using prompts for A and B, respectively, while keeping the input image and model weights fixed. Both predictions are evaluated on the same foreground regions of A and B, and we report median foreground AbsRel over the evaluated pairs.

Figure 4 shows opposite changes in the two objects' errors when the prompt switches from A to B: median foreground AbsRel increases for A and decreases for B, consistently across DA2 and

Table 3: Prompt and region-supervision controls on RLBench and NYU v2. We report median foreground AbsRel for the box and category-name evaluation subsets; lower is better.

Method	RLBench		NYU v2	
	Box	Category-name	Box	Category-name
Full FocusDepth(DA3)	**0.056**	**0.050**	**0.044**	**0.044**
Empty prompt at inference	0.062	0.055	0.046	0.047
Empty prompt during training and inference	0.057	0.053	0.045	0.046
DA3-ft + region loss	0.079	0.074	0.049	0.049
DA3-ft	0.095	0.088	0.046	0.047

Table 4: MSSA component ablations for FocusDepth(DA3) on RLBench. We report median AbsRel across boundary, foreground, and global regions; lower is better.

Method	Box			Category-name		
	Boundary	Foreground	Global	Boundary	Foreground	Global
Full FocusDepth(DA3)	**0.049**	**0.056**	**0.030**	**0.045**	**0.050**	**0.029**
Shuffled spatial correspondence	0.054	0.062	0.033	0.050	0.056	0.033
Shared fusion across scales	0.052	0.060	0.032	0.048	0.055	0.031
Single-MLP fusion	0.052	0.061	0.032	0.048	0.055	0.031
w/o gated injection	0.050	0.059	0.031	0.047	0.051	0.031

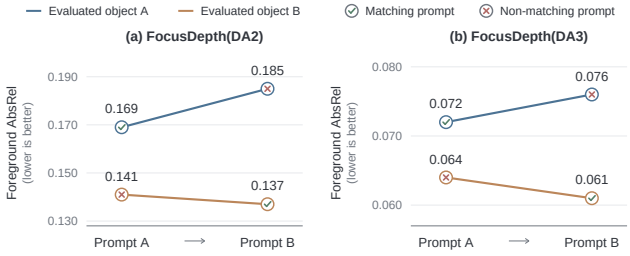


Figure 4: **Same-image target switching on RLBench.** Switching the prompt from A to B increases median foreground AbsRel on A and decreases it on B across both backbones. The panels use different vertical-axis ranges.

Table 5: Training ablations for FocusDepth(DA3) on RLBench. We report median foreground AbsRel; lower is better.

Method	Box	Category-name
FocusDepth(DA3)	**0.056**	**0.050**
Global loss only	0.062	0.056
One-stage training	0.059	0.053

DA3. Prompting the evaluated object rather than another object in the same image reduces median foreground AbsRel by 2.8–8.6% across DA2- and DA3-based variants. These results demonstrate that FocusDepth conditions its refinement on the specified target.

5.4 Ablation Studies

We analyze FocusDepth(DA3) through prompt and region-supervision controls on RLBench and NYU v2, and component and training ablations on RLBench.

Prompt conditioning and region supervision. Table 3 includes two empty-prompt controls: one removes the prompt only at inference, while the other uses empty prompts throughout training and inference with the same architecture, parameter count, losses, two-stage schedule, and optimization budget as Full FocusDepth. Full FocusDepth achieves the lowest foreground AbsRel in all four dataset–prompt settings. The empty-prompt-trained model outperforms the inference-only control but remains less accurate than Full FocusDepth, supporting target-specific prompting under matched capacity and training design. Adding region loss to DA3-ft improves foreground accuracy on RLBench but worsens it on NYU v2. Additional prompt-correctness results are provided in Appendix Section A.4.

MSSA components. In Table 4, shuffling spatial correspondence between prompt-conditioned and geometry tokens causes the largest or joint-largest degradation across all reported metrics. Sharing

fusion across scales or replacing routed fusion with a single MLP worsens foreground and boundary accuracy; removing gated injection produces smaller but consistent increases in error. The shared variant ties the experts, router, and final gate across scales while retaining scale-specific LayerNorm parameters.

Training strategy. Table 5 shows that using only the global depth loss or adopting one-stage training increases foreground AbsRel under both prompt types, supporting region-aware depth supervision and the two-stage training scheme.

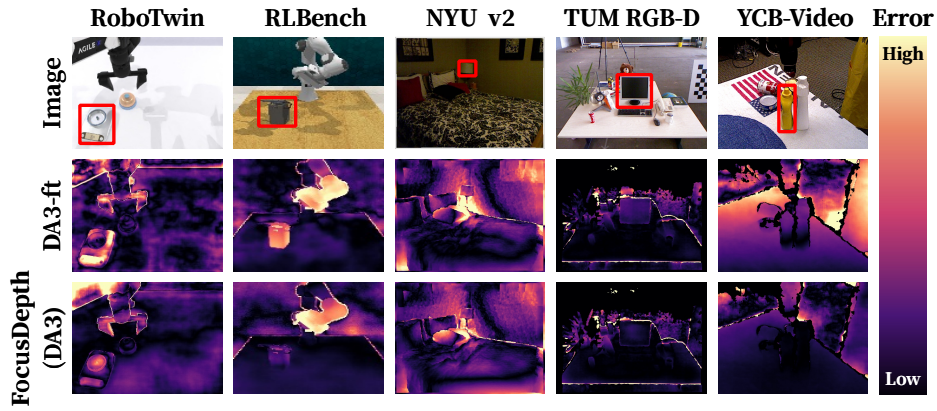


Figure 5: Qualitative comparison of prompt-conditioned depth estimation. Compared with the DA3-ft, FocusDepth(DA3) produces more accurate target-region depth, cleaner boundary transitions, and more coherent local structure around the prompted object or specified target region.

5.5 Qualitative Analysis

Figure 5 shows that DA3-ft can maintain low background errors but still produces pronounced errors on complex prompted objects and their boundaries, visible as highlighted regions in the error maps. With prompt guidance, FocusDepth(DA3) substantially reduces target-region AbsRel, while largely preserving surrounding-scene depth quality. This visual evidence is consistent with the quantitative trend that FocusDepth improves local target depth quality while largely preserving full-image depth accuracy.

6 Conclusion

We introduce Focusable Monocular Depth Estimation (FDE), a target-conditioned formulation of monocular depth estimation that prioritizes foreground accuracy and boundary fidelity around a specified target while retaining full-image relative-depth accuracy. We further propose FocusDepth, a prompt-conditioned monocular relative depth estimation framework that uses Multi-Scale Spatial-Aligned Fusion (MSSA) to inject spatially aligned target cues into dense depth representations for conditional refinement. To make this task measurable, we establish FDE-Bench, which converts diverse RGB-D datasets into standardized image–target–depth triplets and evaluates depth quality over foreground, boundary, and global regions. Experiments on FDE-Bench show that FocusDepth improves target-region depth accuracy and boundary fidelity while retaining full-image relative-depth accuracy, particularly in object-centric and manipulation-oriented scenarios. Together, these results establish FDE as a concrete task for region-prioritized depth estimation, while also indicating areas where the current study remains limited, including larger-scale joint training of FocusDepth and its application to downstream tasks, which can be explored in future work.

Acknowledgments and Disclosure of Funding

This work is funded by NSFC-62676236 and NSFC-62306046.

References

[1] Vasileios Arampatzakis, George Pavlidis, Nikolaos Mitianoudis, and Nikos Papamarkos. Monocular depth estimation: A thorough review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2396–2414, 2023.

[2] Manel Baradad, Yuanzhen Li, Forrester Cole, Michael Rubinstein, Antonio Torralba, William T. Freeman, and Varun Jampani. Background prompting for improved object depth, 2023. URL `https://arxiv.org/abs/2306.05428`.

[3] Amlaan Bhoi. Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*, 2019.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9490–9498. IEEE, 2025.

[6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[7] Pingyi Chen, Yujing Lou, Shen Cao, Jinhui Guo, Lubin Fan, Yue Wu, Lin Yang, Lizhuang Ma, and Jieping Ye. Sd-vlm: Spatial measuring and understanding with depth-encoded vision-language models. *Advances in Neural Information Processing Systems*, 38:33938–33968, 2026.

[8] Souvik Chowdhury and Badal Soni. Envqa: Improving visual question answering model by enriching the visual feature. *Engineering Applications of Artificial Intelligence*, 142:109948, 2025.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

[13] Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast generative monocular depth estimation with flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3203–3211, 2025.

[14] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024.

[15] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[16] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2): 3019–3026, 2020.

[17] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9492–9502, 2024.

[18] Soo Ye Kim, Jianming Zhang, Simon Niklaus, Yifei Fan, Simon Chen, Zhe Lin, and Munchurl Kim. Layered depth refinement with mask guidance, 2022. URL `https://arxiv.org/abs/2206.03048`.

[19] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[20] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17070–17080, 2025.

[21] Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025.

[22] Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. *arXiv preprint arXiv:2511.04555*, 2025.

[23] Sanggyun Ma, Wonjoon Choi, Jihun Park, Jaeyeul Kim, Seunghun Lee, Jiwan Seo, and Sunghoon Im. Bridging geometric and semantic foundation models for generalized monocular depth estimation. In *2026 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–6. IEEE, 2026.

[24] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021.

[25] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the computer vision and pattern recognition conference*, pages 27649–27660, 2025.

[26] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024.

[27] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[28] 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):1623–1637, 2020.

[29] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[30] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[31] Kuanning Wang, Ke Fan, Yuqian Fu, Siyu Lin, Hu Luo, Daniel Seita, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Ocra: Object-centric learning with 3d and tactile priors for human-to-robot action transfer. *arXiv preprint arXiv:2603.14401*, 2026.

[32] Kuanning Wang, Ke Fan, Chenhao Qiu, Zeyu Shangguan, Yuqian Fu, Yanwei Fu, Daniel Seita, and Xiangyang Xue. Oflow: Injecting object-aware temporal flow matching for robust robotic manipulation. *arXiv preprint arXiv:2604.17876*, 2026.

[33] Xinlong Wang, Wei Yin, Tao Kong, Yuning Jiang, Lei Li, and Chunhua Shen. Task-aware monocular depth estimation for 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12257–12264, 2020.

[34] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior, 2025. URL `https://arxiv.org/abs/2505.10565`.

[35] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[36] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024.

[37] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[38] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9043–9053, 2023.

[39] Ziyao Zeng, Daniel Wang, Fengyu Yang, Hyoungseob Park, Yangchao Wu, Stefano Soatto, Byung-Woo Hong, Dong Lao, and Alex Wong. Wordepth: Variational language prior for monocular depth estimation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9708–9719, 2024. URL `https://api.semanticscholar.org/CorpusID:268889577`.

[40] Ziyao Zeng, Yangchao Wu, Hyoungseob Park, Daniel Wang, Fengyu Yang, Stefano Soatto, Dong Lao, Byung-Woo Hong, and Alex Wong. Rsa: Resolving scale ambiguities in monocular depth estimators through language descriptions, 2024. URL `https://arxiv.org/abs/2410.02924`.

[41] Ziyao Zeng, Jingcheng Ni, Daniel Wang, Patrick Rim, Younjoon Chung, Fengyu Yang, Byung-Woo Hong, and Alex Wong. Iris: Integrating language into diffusion-based monocular depth estimation, 2025. URL `https://arxiv.org/abs/2411.16750`.

[42] Deheng Zhang, Yuqian Fu, Runyi Yang, Yang Miao, Tianwen Qian, Xu Zheng, Guolei Sun, Ajad Chhatkuli, Xuanjing Huang, Yu-Gang Jiang, et al. Egonight: Towards egocentric vision understanding at night with a challenging benchmark. *arXiv preprint arXiv:2510.06218*, 2025.

[43] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18537–18546, 2023.

[44] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9): 1612–1627, 2020.

[45] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *2022 international conference on 3D vision (3DV)*, pages 668–678. IEEE, 2022.

A Technical Appendices

A.1 FDE-Bench

A.1.1 FDE-Bench Details

This appendix complements the construction pipeline in Section 4.1 and Figure 3 with dataset statistics, split protocols, and evaluation details for **FDE-Bench**. Table 6 reports the size and prompt availability of each source dataset. Appendix Figures 6–9 document scene diversity, target-scale distributions, regional depth-gradient statistics, and supported prompt formats.

Table 6: Statistics of FDE-Bench, including the images, triplets, prompt types, and object categories.

Dataset	Images		Triplets		Prompt types	Categories
	train	val	train	val		
NYU v2	0.8K	0.7K	22.4K	18.4K	box / text	895
YCB-Video	1.2K	0.1K	7.1K	0.8K	box / text	22
TUM RGB-D	12.5K	1.4K	192.7K	22.3K	box	-
RLBench	1.7K	1.7K	15.1K	15.2K	box / text	55
RoboTwin	4.3K	4.3K	15.6K	15.8K	box	-
FDE-Bench	20.5K	8.2K	252.9K	72.5K	box / text	972

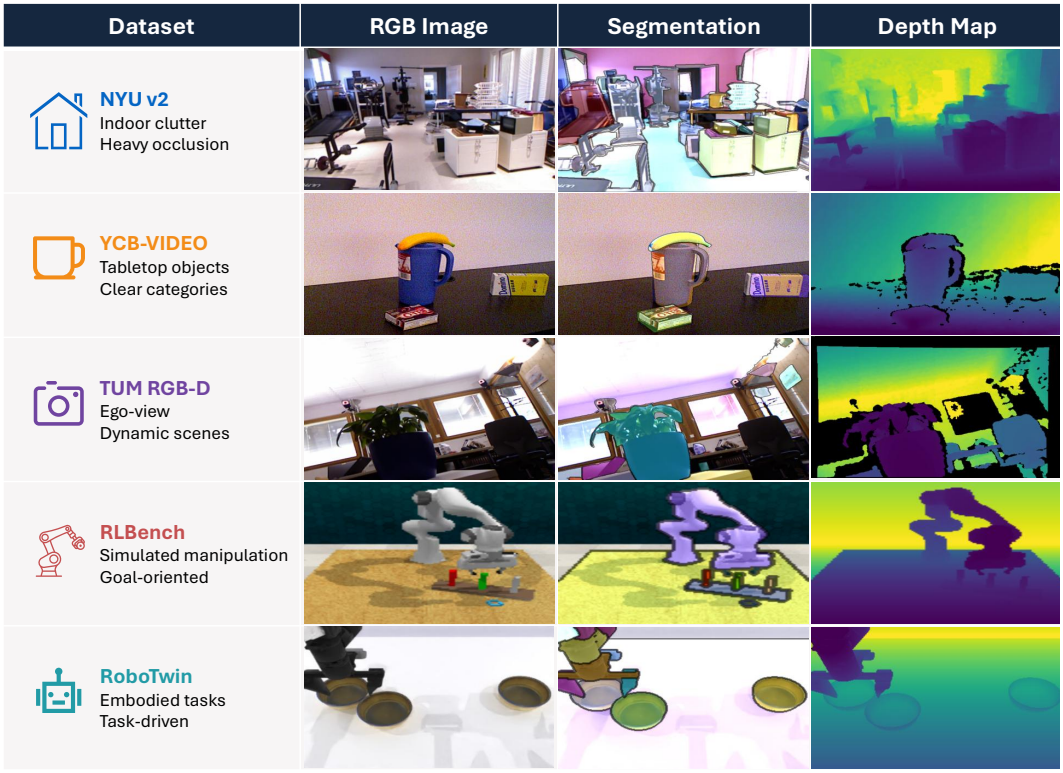


Figure 6: Representative examples from the five source datasets adapted into **FDE-Bench**. For each dataset, we show an RGB image together with its segmentation and depth map, highlighting the diversity of scene structures and target-centric contexts covered by the benchmark, including indoor clutter, large-scale geometry, tabletop objects, ego-view dynamics, and embodied manipulation settings in both real and simulated environments.

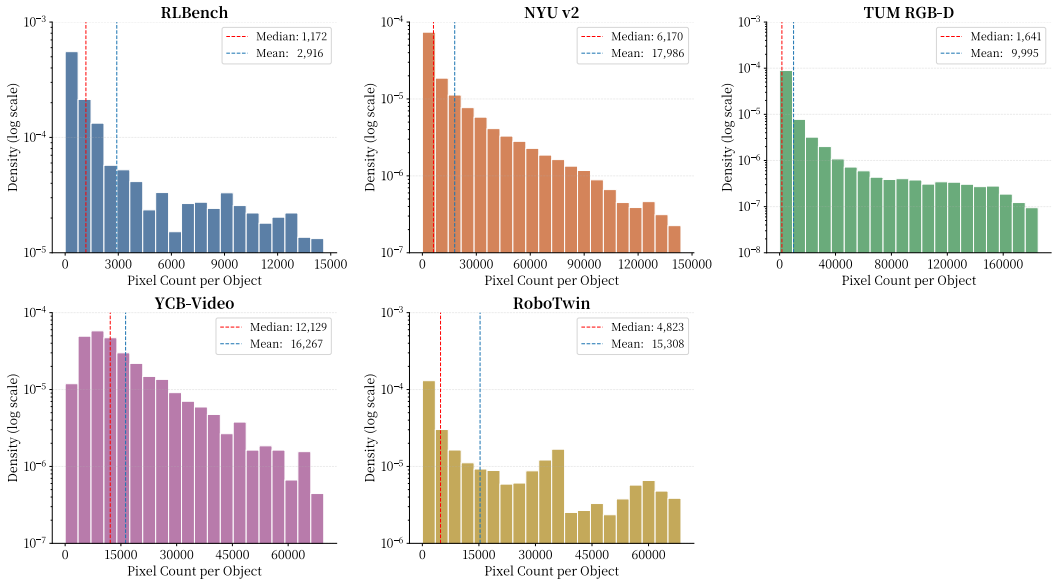


Figure 7: Object size statistics across the source datasets of **FDE-Bench**, measured by pixel count per object. Each subplot shows the distribution of target-object pixel sizes for one adapted dataset, with density plotted on a logarithmic scale and the corresponding mean and median reported. The figure highlights substantial cross-dataset variation in object scale, indicating that FDE-Bench covers targets ranging from small local objects to large scene-dominant instances.

A.1.2 Dataset Scope and Statistics

The five adapted sources listed in Table 6 provide complementary coverage of indoor scenes, tabletop objects, ego-view motion, and manipulation environments. Appendix Figure 6 visualizes representative RGB, segmentation, and depth examples, and Appendix Figure 7 shows that target sizes vary substantially across sources, from small local objects to large scene-dominant regions. Prompt availability follows the annotations of each source: NYU v2, RLBench, and YCB-Video provide both box and category-name prompts, whereas TUM RGB-D and RoboTwin are evaluated with box prompts only, as illustrated in Appendix Figure 9.

All split assignments are made at the image or sequence level to avoid leakage across target triplets. NYU v2 uses its original train/val partition, TUM RGB-D follows the benchmark-specific train/val split, and YCB-Video uses a 10% image-level validation split. For RLBench and RoboTwin, each embodied task contributes two episodes, with one assigned to training and the other to validation.

A.1.3 Region Definitions and Evaluation Protocol

For each triplet, we evaluate three spatial regions: the foreground region, the boundary region, and the global region. Formally, let $D \in \mathbb{R}^{H\times W}, M \in \{0,1\}^{H\times W}, V \in \{0,1\}^{H\times W}$ denote the ground-truth depth map, the target-region mask, and the valid-depth mask. We construct a boundary band B around M with radius $r = 10$ pixels, given by the ring induced by morphological dilation and erosion. The valid foreground, boundary, and global regions are defined as $V_{\text{fg}} = V \cap M, V_{\text{bd}} = V \cap B, V_{\text{glb}} = V$.

For each region, we report $\boldsymbol{\delta_1}$ and **AbsRel**. For relative-depth evaluation, a single scale-shift alignment is estimated over all valid pixels in the full image and then applied unchanged when computing the foreground, boundary, and full-image metrics. Regional metrics are computed per target: each image–target–depth triplet is scored first, and the resulting target-level values are then aggregated using the mean or median, as specified in the corresponding experimental results.

Appendix Figure 8 provides supporting statistics by plotting depth-gradient distributions for foreground, boundary, and background regions across the source datasets. Boundary regions generally

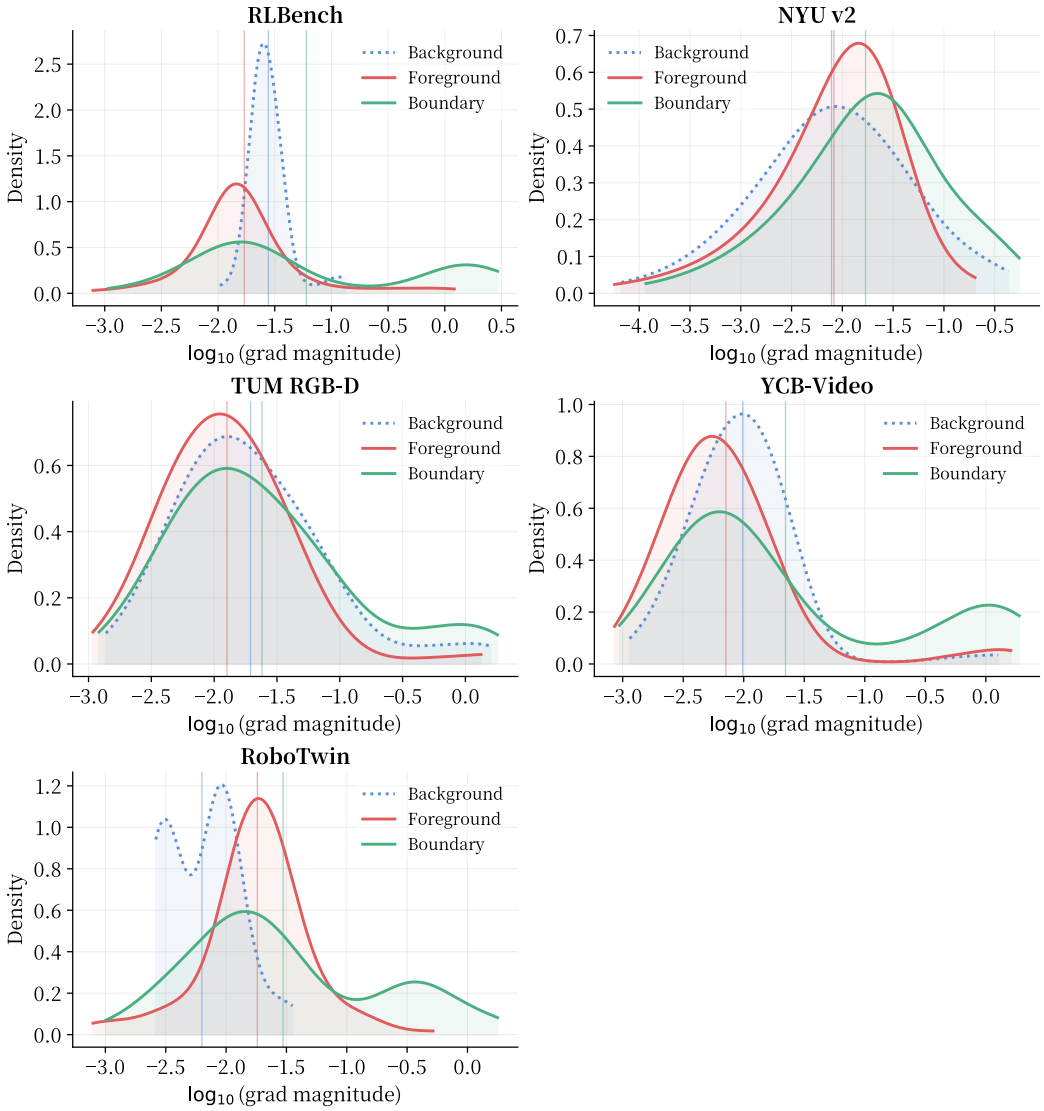


Figure 8: Depth-gradient statistics across the source datasets of **FDE-Bench**. For each dataset, we plot the distribution of depth gradient magnitudes, shown as $\log_{10}$(grad magnitude), separately for background, foreground, and boundary regions. The distributions reveal systematic regional differences, with boundary areas generally exhibiting stronger depth variations, supporting the region-aware design of FDE evaluation.

exhibit stronger depth variation, which motivates reporting boundary metrics separately from foreground and whole-image metrics.

A.2 Training Hyperparameters

Tables 7 and 8 report the training hyperparameters used for FocusDepth and the fine-tuned DA2/DA3 baselines, respectively. FocusDepth uses a two-stage schedule over the same five-epoch training budget, whereas the DA2-ft and DA3-ft baselines are trained with a single-stage fine-tuning schedule.

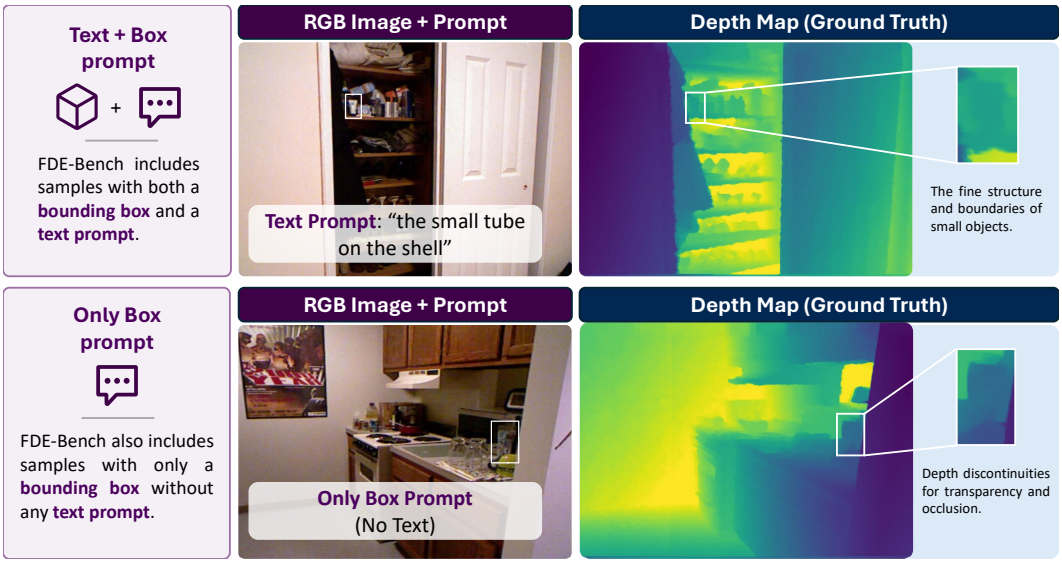


Figure 9: Illustration of target specification formats in **FDE-Bench**. Depending on the adapted source dataset, a target may be specified by a bounding box together with a category-name prompt, or by a bounding box alone. The examples also highlight typical target-centric depth challenges addressed by FDE, including fine object structures, boundary fidelity, and depth discontinuities caused by occlusion or transparency.

Table 7: Training hyperparameters for FocusDepth.

Hyperparameter	**Stage 1**	**Stage 2**
Learning rate	1×10^{-4}	1×10^{-5}
Batch size per-device	4	4
Input resolution	1008×1008	1008×1008
Max steps	1 epoch	4 epochs
Warmup steps	1/10 of Stage 1 max steps	1/10 of Stage 2 max steps
LR scheduler	Cosine decay with linear warmup	Cosine decay with linear warmup
Weight decay	1×10^{-5}	1×10^{-5}
Dropout	0.1	0.1
Optimizer	AdamW	AdamW
Gradient clipping	1.0	1.0

A.3 Additional Experimental Results

Table 2 reports the text-prompt comparison on NYU v2, RLBench, and YCB-Video, complementing the main box-prompt results with zero-shot baselines and fine-tuned DA2/DA3 variants.

Table 9 reports the box-prompt comparison on TUM RGB-D and YCB-Video, where zero-shot baselines are shown together with DA3 fine-tuning variants for the datasets not included in the main box-prompt table.

A.4 RLBench Prompt-Correctness Study

We conduct this study to examine the sensitivity of FocusDepth(DA3) to prompt correctness in the FDE task. The evaluation includes a correct target prompt, a wrong prompt referring to another visible object, a wrong prompt referring to an absent object, and an empty prompt. Unlike the main comparison, which reports medians, this study reports means to better expose the aggregate effect of bad cases under prompt perturbations; the corresponding mean–median comparison is provided in Appendix Table 10.

Table 8: Training hyperparameters for the DA2-ft and DA3-ft baselines.

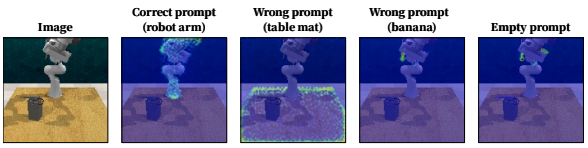
Hyperparameter	**Value**
Learning rate	1×10^{-5}
Batch size per-device	4
Input resolution	1008×1008
Max steps	5 training epochs
Warmup steps	1/10 of max steps
LR scheduler	Cosine decay with linear warmup
Weight decay	1×10^{-5}
Dropout	0.1
Optimizer	AdamW
Gradient clipping	1.0

Table 9: Additional box-prompt results on TUM RGB-D and YCB-Video. We compare zero-shot depth foundation models, the fine-tuned DA3 baseline, and FocusDepth variants across boundary, foreground, and global regions.

Dataset	**Method**	**Boundary region**		**Foreground region**		**Global region**	
		$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$
TUM RGB-D [30]	UniDepth-v1 [26]	0.980 (0.744, 1.000)	0.092	**1.000** (0.867, 1.000)	0.078	0.948 (0.845, 0.985)	0.085
	UniDepth-v2 [27]	0.979 (0.805, 1.000)	**0.080**	**1.000** (**0.922**, 1.000)	**0.061**	**0.967** (**0.877**, 0.987)	**0.070**
	Metric3D-v2 [14]	**0.984** (**0.821**, 1.000)	0.087	**1.000** (0.913, 1.000)	0.076	0.957 (0.856, **0.989**)	0.083
	DA2 [37]	0.960 (0.709, 1.000)	0.095	0.999 (0.818, 1.000)	0.080	0.946 (0.849, 0.983)	0.076
	DA3 [19]	0.964 (0.699, 1.000)	0.099	0.999 (0.790, 1.000)	0.080	0.938 (0.778, 0.980)	0.093
	DA3-ft	0.999 (0.853, 1.000)	0.068	**1.000** (0.944, 1.000)	0.057	0.970 (0.905, 0.990)	0.060
	FocusDepth(DA3)	**1.000** (**0.882**, 1.000)	**0.061**	**1.000** (**0.966**, 1.000)	**0.051**	**0.975** (**0.912**, **0.991**)	**0.057**
YCB-Video [35]	UniDepth-v1 [26]	**1.000** (0.984, 1.000)	0.042	1.000 (**0.998**, 1.000)	0.037	0.993 (0.981, 0.997)	0.045
	UniDepth-v2 [27]	0.999 (0.965, 1.000)	**0.036**	1.000 (0.994, 1.000)	**0.026**	0.996 (**0.991**, 0.998)	**0.031**
	Metric3D-v2 [14]	**1.000** (**0.986**, 1.000)	0.040	1.000 (**0.998**, 1.000)	0.031	**0.997** (0.988, **0.999**)	0.042
	DA2 [37]	0.999 (0.963, 1.000)	0.045	1.000 (0.993, 1.000)	0.038	0.994 (0.972, 0.998)	0.040
	DA3 [19]	0.994 (0.938, 1.000)	0.054	0.999 (0.986, 1.000)	0.045	0.992 (0.961, 0.997)	0.047
	DA3-ft	**1.000** (0.988, 1.000)	0.042	**1.000** (**0.999**, 1.000)	0.036	0.997 (0.984, **0.999**)	0.044
	FocusDepth(DA3)	**1.000** (**0.990**, 1.000)	**0.037**	**1.000** (**0.999**, 1.000)	**0.026**	**0.998** (**0.992**, **0.999**)	**0.032**

As shown in Appendix Figure 10, correct prompts produce the strongest target-centric response and the best boundary and foreground results. Degraded prompts mainly reduce local target-region gains, with smaller effects on global accuracy. Notably, even wrong or empty prompts remain stronger than the DA3-ft no-prompt baseline across boundary, foreground, and global regions.

(a) Prompt-conditioned attention visualization.



(b) Quantitative prompt-correctness study with mean values.

Prompt Setting	Boundary region		Foreground region		Global region	
	$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$
Correct	**0.919**	**0.108**	**0.901**	**0.137**	**0.943**	0.075
Wrong visible	0.914	0.111	0.884	0.148	**0.943**	**0.074**
Wrong absent	0.911	0.117	0.886	0.152	0.939	0.078
Empty	0.910	0.118	0.886	0.154	0.939	0.078
DA3-ft	0.860	0.162	0.804	0.226	0.915	0.105

Figure 10: RLBench prompt-correctness study for FocusDepth(DA3) under category-name prompts. (a) Prompt-conditioned attention visualization under correct, wrong-visible, absent, and empty prompts. (b) Quantitative results reported as means; the corresponding mean–median comparison is provided in Appendix Table 10. Best results are shown in bold.

A.5 RLBench Prompt-Correctness Mean–Median Comparison

The prompt-correctness study in Section A.4 reports mean values, unlike the main comparative results that report medians. We use means in this diagnostic experiment because prompt perturbations can create bad cases whose aggregate effect is important for measuring robustness. Table 10 reports both means and medians for the same RLBench category-name-prompt setting, showing how the diagnostic mean values relate to the median-based protocol used in the main comparison.

Table 10: Mean–median comparison for the RLBench prompt-correctness study under category-name prompts. Mean values are reported for this diagnostic study to better reflect bad cases under prompt perturbations; medians are included here for comparison with the main evaluation metrics.

Method	Boundary region				Foreground region				Global region			
	δ_1 mean ↑	δ_1 med. ↑	AbsRel mean ↓	AbsRel med. ↓	δ_1 mean ↑	δ_1 med. ↑	AbsRel mean ↓	AbsRel med. ↓	δ_1 mean ↑	δ_1 med. ↑	AbsRel mean ↓	AbsRel med. ↓
Correct prompt	**0.919**	**0.995**	**0.108**	**0.045**	**0.901**	**1.000**	**0.137**	**0.050**	**0.943**	0.996	0.075	**0.029**
Wrong prompt (other visible object)	0.914	0.995	0.111	0.045	0.884	**1.000**	0.148	0.052	**0.943**	**0.996**	**0.074**	**0.029**
Wrong prompt (absent object)	0.911	0.994	0.117	0.047	0.886	**1.000**	0.152	0.053	0.939	0.995	0.078	0.030
Empty prompt	0.910	0.993	0.118	0.045	0.886	**1.000**	0.154	0.055	0.939	0.995	0.078	**0.029**
DA3-ft no-prompt baseline	0.860	0.973	0.162	0.068	0.804	0.998	0.226	0.088	0.915	0.984	0.105	0.041

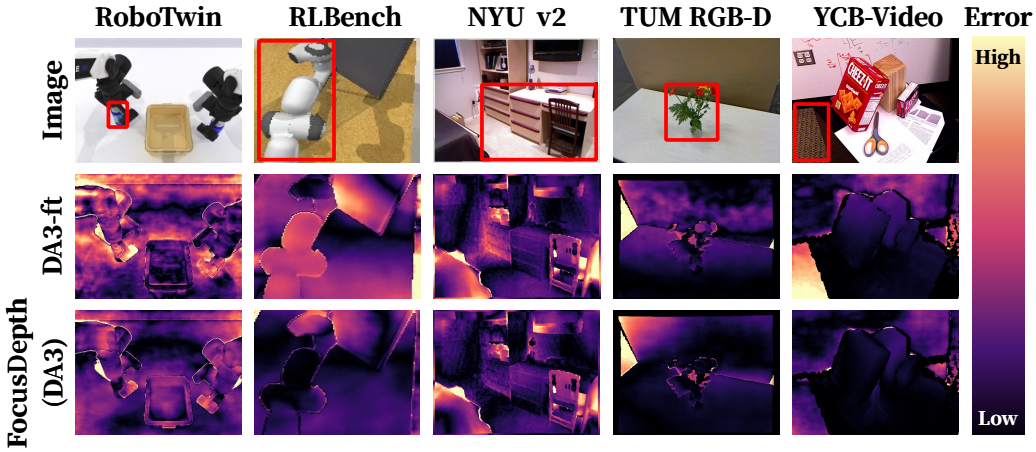


Figure 11: Additional qualitative comparison of prompt-conditioned depth estimation. Compared with the DA3-ft, FocusDepth(DA3) produces more accurate target-region depth, cleaner boundary transitions, and more coherent local structure around the prompted object or specified target region.

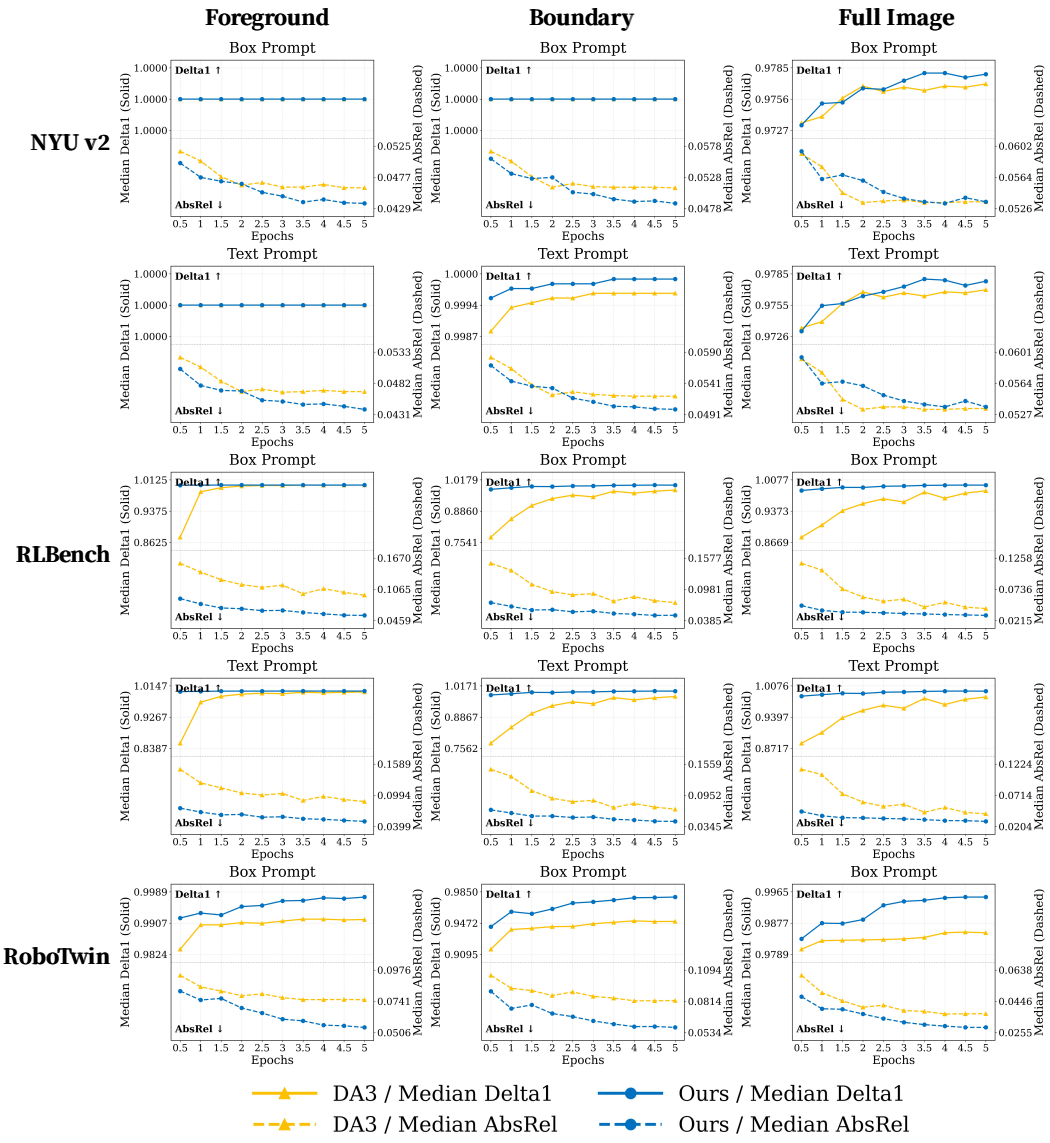


Figure 12: Validation curves of DA3-ft and FocusDepth(DA3) over training epochs. We compare δ_1 and AbsRel on NYU v2, RLBench, and RoboTwin across foreground, boundary, and global regions. FocusDepth(DA3) consistently improves target-relevant regions, especially foreground and boundary areas, while preserving global depth quality.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the proposed work in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not include theoretical results. Therefore, the requirement to provide assumptions and complete proofs is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the information needed to reproduce the main experimental results in the method and experimental sections. These sections describe the proposed approach, experimental setup, implementation details, and evaluation metrics relevant to the main claims and conclusions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide reproducible code with sufficient instructions in the supplemental material. The submitted materials will include the necessary setup information and commands to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the necessary training and testing details in the experiment and appendix sections, including the experimental setup, data splits, hyperparameters, and optimization settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the median, 25th percentile, and 75th percentile for the main experimental results. These quantile-based statistics summarize the distribution of the results and the 25th–75th percentile range captures the variability across evaluation instance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the computational resources used for the experiments in the experiment section and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics. The experiments do not involve human subjects, sensitive personal data, or other apparent ethical concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This work is foundational research and is not directly tied to a specific application or deployment scenario. Therefore, it does not present an immediate or direct societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper does not release data or models that pose a high risk of misuse or dual-use concerns.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The data and models used in this paper are publicly available open-source assets. We properly credit their original creators and respect the corresponding licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper introduces new assets, including a new framework and a new benchmark. We provide documentation for these assets, including their intended use, evaluation protocol, limitations, and relevant implementation details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing experiments or research with human subjects; therefore, IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.