SUPPLEMENTARY MATERIALS OF LOTUS: DIFFUSION-BASED VISUAL FOUNDATION MODEL FOR HIGH-QUALITY DENSE PREDICTION

Anonymous authors
Paper under double-blind review

A EXPERIMENTAL SETTINGS

A.1 IMPLEMENTATION DETAILS

We implement Lotus based on Stable Diffusion V2 (Rombach et al., 2022), with text conditioning disabled. During training, we fix the time-step $t = 1000$. To optimize the model, we utilize the standard Adam optimizer with the learning rate 3×10^{-5}. All experiments are conducted on 8 NVIDIA A800 GPUs and the total batch size is 128. For our discriminative variant, we train for 4,000 steps, which takes $\sim$8.1 hours, while for the generative variant, we extend training to 10,000 steps, requiring $\sim$20.3 hours.

A.2 EVALUATION DATASETS AND METRICS

Evaluation Datasets. ① For affine-invariant depth estimation, we evaluate on 4 real-world datasets that are not seen during training: NYUv2 (Silberman et al., 2012) and ScanNet (Dai et al., 2017) all contain images of indoor scenes; KITTI (Geiger et al., 2013) contains various outdoor scenes; ETH3D (Schops et al., 2017), a high-resolution dataset, containing both indoor and outdoor scenes. ② For surface normal prediction, we employ 4 datasets for evaluation: NYUv2 (Silberman et al., 2012), ScanNet (Dai et al., 2017), and iBims-1 (Koch et al., 2018) contain real indoor scenes; Sintel (Butler et al., 2012) contains highly dynamic outdoor scenes.

Metrics. ① For affine-invariant depth, we follow the evaluation protocol from (Ranftl et al., 2020; Ke et al., 2024; Yang et al., 2024a;b), aligning the estimated depth predictions with available ground truths using least-squares fitting. The accuracy of the aligned predictions is assessed using the *absolute mean relative error* (AbsRel), *i.e.*, $\frac{1}{M}\sum_{i=1}^{M}|a_i - d_i|/d_i$, where M is the total number of pixels, a_i is the predicted depth map and d_i represents the ground truth. We also report $\delta 1$ and $\delta 2$, the proportion of pixels satisfying $\text{Max}(a_i/d_i, d_i/a_i) < 1.25$ and $< 1.25^2$ respectively.

② For surface normal, following (Bae & Davison, 2024; Ye et al., 2024), we evaluate the predictions of Lotus by measuring the mean angular error for pixels with available ground truth. Additionally, we report the percentage of pixels with an angular error below 11.25° and 30°.

For all tasks, we report the *Avg. Rank*, which indicates the average ranking of each method across various datasets and evaluation metrics. A lower value signifies better overall performance.

B DETAILS OF DIRECT ADAPTION

As illustrated in Fig. 4 of the main paper, our Direct Adaption means directly adapting the standard diffusion formulation for dense prediction task with minimal modifications. Specifically, starting with the pre-trained Stable Diffusion model, image $\mathbf{x}$ and annotation $\mathbf{y}$ are encoded using the pre-trained VAE encoder. Noise is added to the encoded annotation to obtain the noisy annotation $\mathbf{z}_\mathbf{t}^\mathbf{y}$ at noise level $t \in [1, T]$. The encoded image $\mathbf{z}^\mathbf{x}$ is then concatenated with the noisy annotation $\mathbf{z}_\mathbf{t}^\mathbf{y}$ to form the input of the denoiser U-Net model. To handle this concatenated input, the U-Net input layer is duplicated (from 4 channels to 8 channels) and its original weights are halved as initialization, which prevents activation inflation (Ke et al., 2024). Direct Adaptation is optimized using the standard multi-step formulation the standard diffusion objective, ϵ-prediction, as described in Eq. 2 of the main paper. To analyze the original diffusion formulation more effectively, we avoid

specialized techniques introduced in prior methods (Ke et al., 2024; Fu et al., 2024; Xu et al., 2024; Ye et al., 2024), such as annealed multi-resolution noise and test-time ensembling.

C Analysis of "direction($\mathbf{z}_\tau^\mathbf{y}$)" in DDIM Process (Eq. 4)

In addition to the predicted clean sample $\hat{\mathbf{z}}_\tau^\mathbf{y}$, Eq. 4 of the main paper includes another term, "direction($\mathbf{z}_\tau^\mathbf{y}$)". It is calculated according to different parameterization types:

$$\begin{aligned} &\epsilon\text{-prediction: } d = w_\tau \cdot f_\theta^\epsilon \\ &x_0\text{-prediction: } d = w_\tau \cdot [\frac{1}{\sqrt{1-\overline{\alpha}_\tau}}(\mathbf{z}_\tau^\mathbf{y} - \sqrt{\overline{\alpha}_\tau} f_\theta^\mathbf{z})] \end{aligned} \tag{A}$$

where d represents the term "direction($\mathbf{z}_\tau^\mathbf{y}$)", $w_\tau = \sqrt{1-\overline{\alpha}_{\tau-1}}$ is the weight at denoising step τ. And f_θ^ϵ and $f_\theta^\mathbf{z}$ denote the model outputs for different parameterizations. For clarity, the input of the model f_θ is omitted. As shown in Eq. A, for x_0-prediction, when $\tau \to 1$, *i.e.*, at the end of the denoising process, the factor $\sqrt{1-\overline{\alpha}_\tau} \to 0$, which may amplify variance from $f_\theta^\mathbf{z}$. However, its influence is limited. The reasons are as follows: ① The rate of change of $\sqrt{1-\overline{\alpha}_\tau}$ from T to 1 is initially slow and then accelerates. As a result, the factor remains close to 1 for most of the denoising process, only close to 0 in the final steps. ② In x_0-prediction, compared to the initial denoising steps, the gap between network output $f_\theta^\mathbf{z}$ and $\mathbf{z}_\tau^\mathbf{y}$ in the final steps is much weaker and gradually approaching zero. With $\sqrt{\overline{\alpha}_\tau} \to 1$ as $\tau \to 1$, we can get $\mathbf{z}_\tau^\mathbf{y} - \sqrt{\overline{\alpha}_\tau} f_\theta^\mathbf{z} \to 0$, which may also indicate the limited influence of factor $\sqrt{1-\overline{\alpha}_\tau}$.

D Performance of v-prediction

In sec. 4.1, we discussed two basic parameterization types: ϵ-prediction and x_0-prediction. The latest parameterization, v-prediction (Salimans & Ho, 2022), combines these two basic parameterizations to avoid the invalid prediction values of ϵ-prediction at some timesteps for progressive distillation. Specifically, the U-Net denoiser model f_θ learns to predict the combination of added noise ϵ and the clean sample $\mathbf{z}^\mathbf{y}$: $\mathbf{v} = \sqrt{\overline{\alpha}_\tau}\epsilon - \sqrt{1-\overline{\alpha}_\tau}\mathbf{z}^\mathbf{y}$, where $\sqrt{\overline{\alpha}_\tau}^2 + \sqrt{1-\overline{\alpha}_\tau}^2 = 1$. During inference, according to the Eq. 4 of main paper, the prediction $\hat{\mathbf{z}}_\tau^\mathbf{y} = \sqrt{\overline{\alpha}_\tau}\mathbf{z}_\tau^\mathbf{y} - \sqrt{1-\overline{\alpha}_\tau} f_\theta^\mathbf{v}$, where $f_\theta^\mathbf{v}$ represents the predicted combination, striking a balance between ϵ (ϵ-prediction) and $\mathbf{z}^\mathbf{y}$ (x_0-prediction). As shown in Fig. A, we conduct experiments based on the settings in Fig. 5 and 6 of the main paper. The results indicate that the performance of v-prediction falls between that of x_0-prediction and ϵ-prediction, with moderate variance. However, for dense prediction tasks, minimizing variance is crucial to avoid unstable prediction. Therefore, v-prediction may not be the optimal choice. In contrast, x_0-prediction achieves the best performance with the lowest variance, which is why we replace the standard ϵ-prediction with the more suitable x_0-prediction.

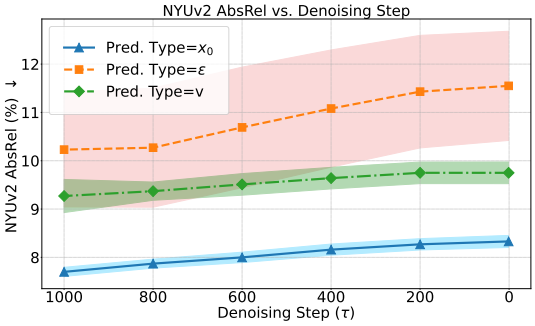


Figure A: **Quantitative evaluation of the predicted depth maps $\hat{\mathbf{z}}_\tau^\mathbf{y}$ along the denoising process.** The experimental settings are same as Fig. 5 and 6. Six steps are selected for illustration. The banded regions around each line indicate the variance, wider areas representing larger variance.

E Qualitative Comparisons

In Fig. B, we further compare the performance of our Lotus with other methods in detailed areas. The quantitative results obviously demonstrate that our method can produce much finer and more accurate depth predictions, particularly in complex regions with intricate structures, which sometimes cannot be reflected by the metrics. Also, as illustrated in Fig. C, Lotus consistently provides accurate surface normal predictions, effectively handling complex geometries and diverse environments, highlighting its robustness on fine-grained prediction.

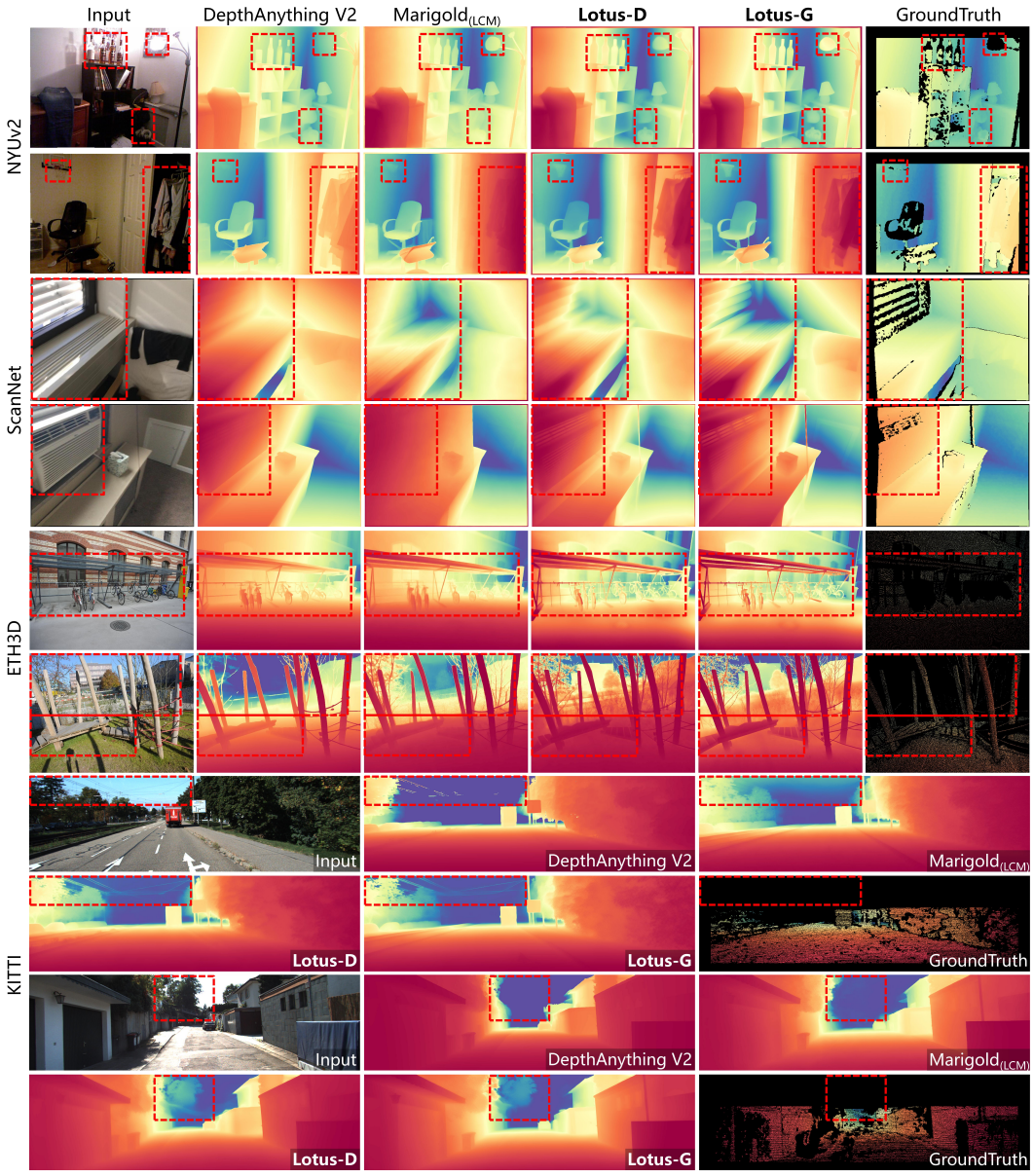


Figure B: **Qualitative comparison on zero-shot affine-invariant depth estimation.** Lotus demonstrates higher accuracy especially in detailed areas.

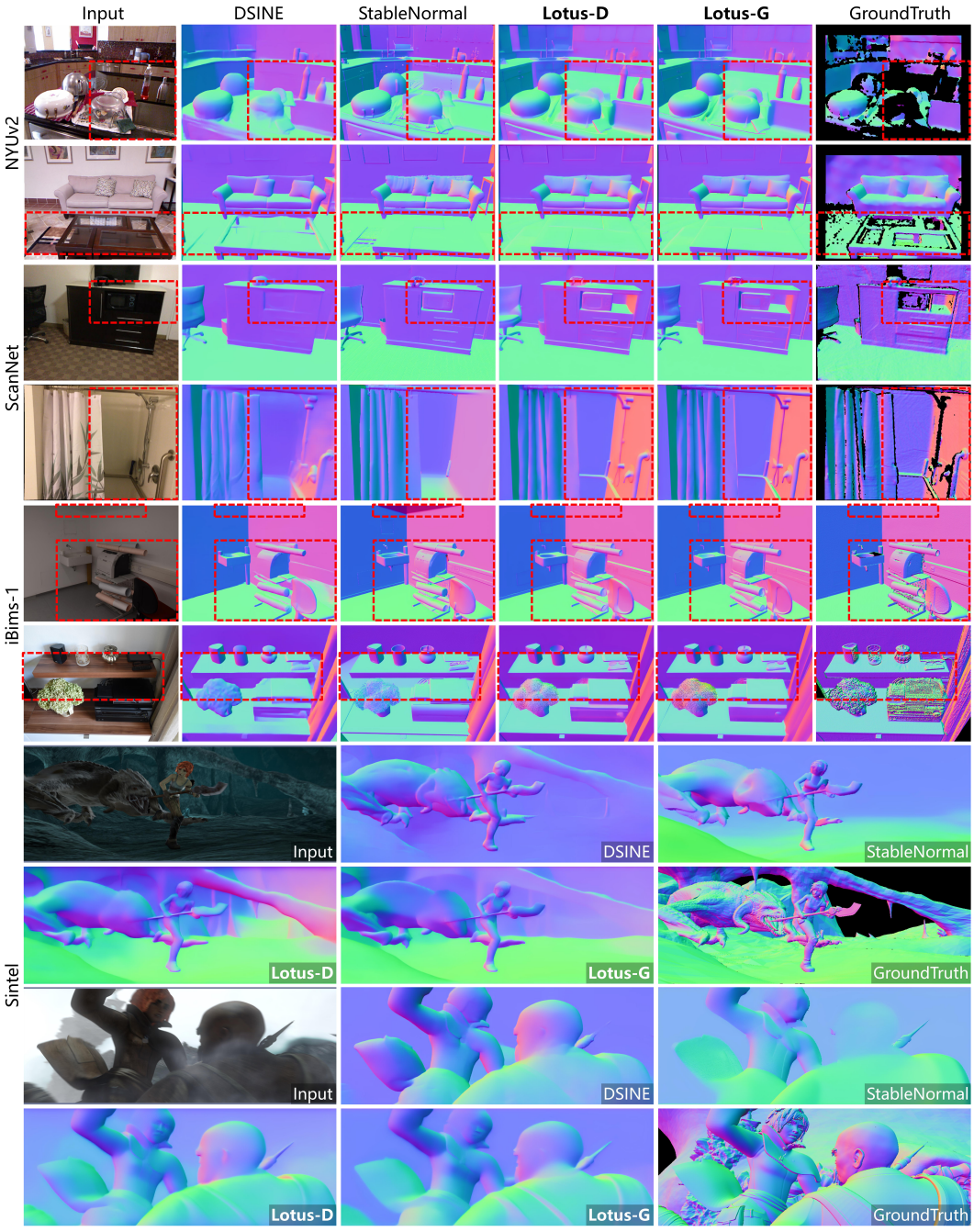


Figure C: **Qualitative comparison on zero-shot surface normal estimation.** Lotus offers improved accuracy particularly in complex regions.

F EXPERIMENTS ON MORE DENSE PREDICTION TASKS: SEMANTIC SEGMENTATION AND DIFFUSE REFLECTANCE

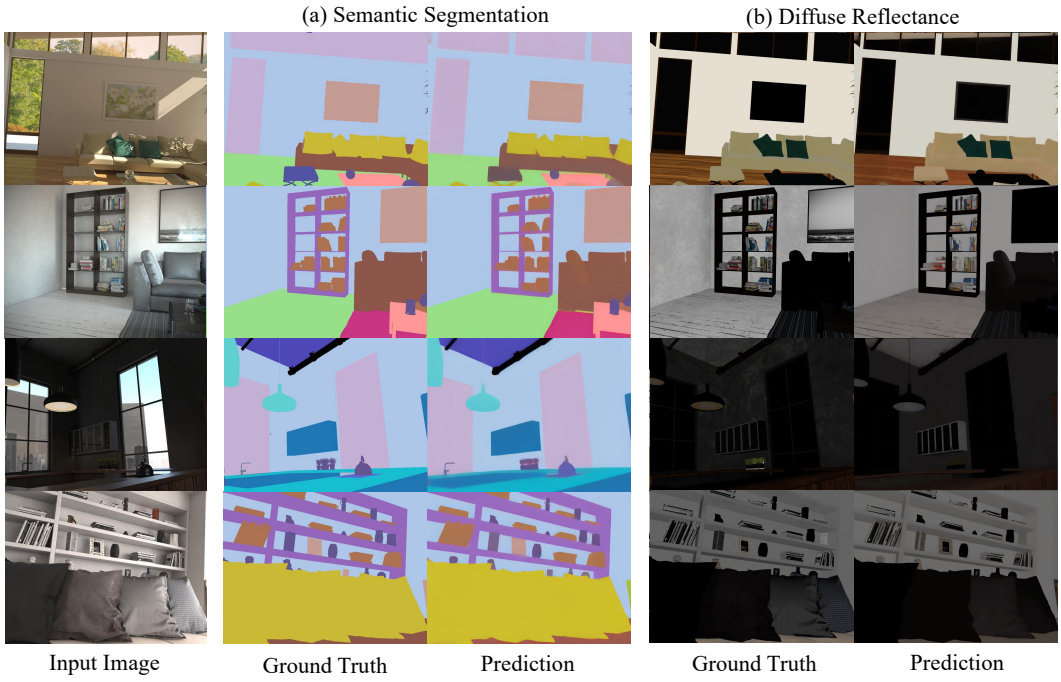


Figure D: **Experiments of Lotus on (a) semantic segmentation and (b) diffuse reflectance.** The high-quality results indicate that our method, even without task-specific designs, can be effectively applied not only to geometric dense prediction tasks, but also to semantic dense prediction tasks.

Table A: The quantitative results of semantic segmentation on Hypersim (Roberts et al., 2021) testing set. Mean values are reported from 10 independent runs.

Method	mIoU ↑	mAcc ↑
Direct Adaption	14.1	61.3
Lotus-G	**21.2**	**65.6**

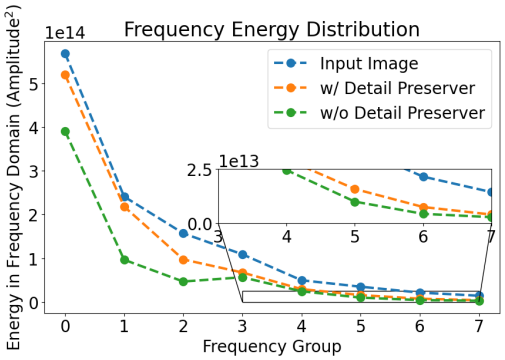
Table B: The quantitative results of diffuse reflectance prediction on Hypersim (Roberts et al., 2021) testing set. Mean values are reported from 10 independent runs.

Method	L1 ↓	L2 ↓
Direct Adaption	0.198	0.206
Lotus-G	**0.109**	**0.135**

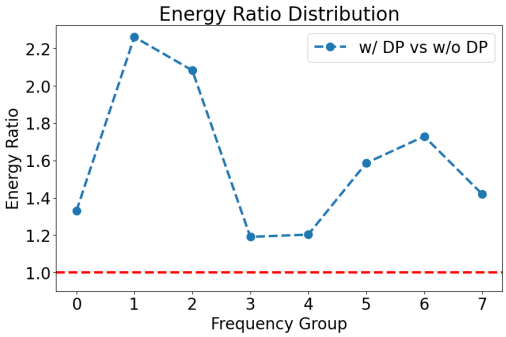
To validate the generalization ability of our method on other dense prediction tasks, we further train it on semantic segmentation and diffuse reflectance prediction. Both tasks are trained using the training set of the Hypersim dataset (Roberts et al., 2021) and evaluated on their corresponding test sets. For semantic segmentation, we report the mean intersection over union (mIoU) and mean accuracy (mAcc). For diffuse reflectance prediction, we evaluate using the L1 and L2 distances to the ground truth. To enable fast evaluation, we randomly select 500 paired testing samples. In our experiments, we do not redesign any specific modules or loss functions for these tasks and maintain the original training protocol of Lotus unchanged. As shown in Tab. A and Tab. B, we compare our method with the baseline, Direct Adaption (Fig. 4 in the main paper), to assess its effectiveness. The results show that our method outperforms the baseline across all metrics. Additionally, we provide qualitative visualizations for these two tasks in Fig. D, demonstrating accurate and high-quality results. Both the quantitative and qualitative results indicate that our method, even without task-specific designs, can be effectively applied not only to geometric dense prediction tasks, as shown in the main paper, but also to semantic dense prediction tasks.

G Frequency Domain Analysis of the Detail Preserver Take Monocular Depth Estimation as An Example

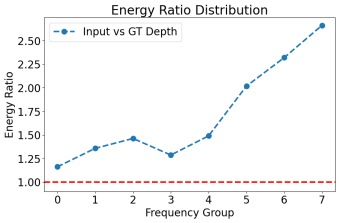
We use fast Fourier transform (FFT) to compute the Discrete Fourier Transform (DFT) of the input images and depth map estimations with and without Detail Preserver. The entire 2D frequency domains are divided into 8 frequency groups exponentially using the base of 2, *i.e.*, the first group covers the 2D frequency map in a circle with a radius of 2, the second group covers the annular region with radii from 2 to 4, the third group covers radii from 4 to 8, and so on. This exponential grouping allows us to analyze the frequency components across progressively larger ranges, capturing both low-frequency and high-frequency characteristics.



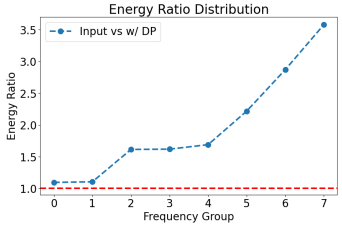
(a) Frequency domain energy distribution comparisons among input image, and depth estimations *w/* and *w/o* Detail Preserver.



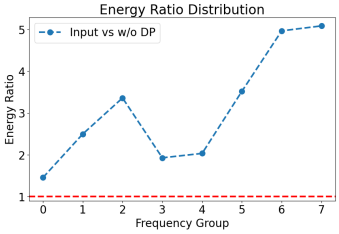
(b) Frequency domain energy ratio between the depth estimations *w/* and *w/o* Detail Preserver.



(c) Frequency energy ratio **between input image and GT depth.**



(d) Frequency energy ratio **between input image and depth estimations *w/* Detail Preserver.**



(e) Frequency domain energy ratio **between the input image and depth estimations *w/o* Detail Preserver.**

Figure E: **Frequency Domain Analysis of the Detail Preserver** We use Hypersim (Roberts et al., 2021) dataset to transfer the input image and depth estimation *w/* and *w/o* Detail Preserver into 2D frequency domains, using FFT. 100 pairs of {input image, depth estimation *w/* Detail Preserver, depth estimation *w/o* Detail Preserver} are randomly selected for this frequency domain analysis. Hypersim is a photorealistic synthetic dataset. Not only can Hypersim offer dense GT labels without `None` areas (which is important during FFT), its depth annotations are much fine-grained compared with real-world datasets like NYUv2 (Silberman et al., 2012) and KITTI Geiger et al. (2013).

In order to more clearly demonstrate the effect of our proposed Detail Preserver, we first analysis the experiments using Hypersim (Roberts et al., 2021) dataset to display the difference in frequency domain energy between the details from both geometry and texture (the input images); and the details from purely the geometry (the GT depth maps). As shown in Fig. Ec, the frequency domain energy between the input images and the depth annotations are plotted. Clearly we can see that the input images has much higher frequency energy in high-frequency areas, *i.e.*, group 4, 5, 6, and 7, indicating that the details in surface textures mainly contribute to high-frequency energy; while the details in geometries, which can be expressed by depth maps, are mainly concentrated into (relative) middle and low frequency areas, *i.e.*, group 0, 1, 2, and 3.

As shown in Fig. Ea and Eb, collaborating with the Detail Preserver effectively drag the frequency domain energy of depth estimation to the input image, especially on middle and low frequency domains, *i.e.*, the frequency group 0, 1, 2 and 3, highlighting the Detail Preserver's effectiveness in enhancing the geometrical details that should be reflected into depth predictions, like the fences around roads and houses (Fig. 8 of our main paper).While for high-frequency components, *i.e.*, the frequency group 4, 5, 6, and 7, which may be primarily caused by the highly detailed textures, like the signs on the road and patterns on house surfaces, the energy in these areas between depth estimations with and without Detail Preserver is quite similar, indicating that the Detail Preserver does not copy this high-frequency and geometry-independent texture.

By comparing Fig. Ec, Ed and Ee together, we can see that Detail Preserver effectively enhances the details of geometries. This insight is evident by this phenomenon: the frequency domain energy ratio between input and depth estimation *w/* Detail Preserver, is closer to the frequency domain energy ratio between input and GT depth, compared with the frequency domain energy ratio between input and depth estimation *w/o* Detail Preserver.

H Evaluate the Lotus-G Given Randomness & Lotus's Performance on DIODE and OASIS

To ensure a fairer comparison, we re-evaluate the performance of Lotus-G given 10 independent runs under different seeds, the metric values are reported in "mean (±std)" format. Please see Tab. C for the Lotus-G's results in depth estimation, using true-depth space. Please see Tab. D for the Lotus-G's results in depth estimation, using disparity space, which delivers better results compared with true-depth space, becoming the new SoTA in diffusion-based depth estimation. Please see Tab. G for the Lotus-G's results in normal estimation.

Table C: **Lotus-G's results in zero-shot affine-invariant depth estimation, using true-depth space.** 10 independent runs are conducted for calculating the mean and std value. The results on DIODE is also reported.

Dataset	AbsRel↓	RMSE↓	$\delta1$↑	$\delta2$↑	$\delta3$↑
DIODE	0.3311 (±1.4285e-04)	3.8836 (±1.3504e-03)	0.7360 (±4.4283e-04)	0.8764 (±5.4613e-04)	0.9304 (±1.4050e-04)
ETH3D	0.0617 (±1.9343e-04)	0.5806 (±9.8765e-04)	0.9605 (±4.3380e-04)	0.9897 (±8.3054e-05)	0.9957 (±7.8772e-05)
KITTI	0.1134 (±5.8194e-05)	3.5379 (±8.9554e-04)	0.8771 (±1.3912e-04)	0.9776 (±6.6427e-05)	0.9930 (±4.0593e-05)
NYUv2	0.0542 (±1.0072e-04)	0.2220 (±1.5572e-04)	0.9661 (±2.5504e-04)	0.9915 (±7.9410e-05)	0.9978 (±2.9075e-05)
ScanNet	0.0603 (±1.7740e-04)	0.1597 (±3.5362e-04)	0.9590 (±1.4926e-04)	0.9893 (±2.3924e-04)	0.9972 (±1.0019e-04)

Table D: **Lotus-G's results in zero-shot affine-invariant depth estimation, using disparity space,** following DepthAnything (Yang et al., 2024a;b) series. 10 independent runs are conducted for calculating the mean and std value. Compared with Lotus-G in true-depth space, training in disparity further enhances the performance. The results on DIODE is also reported.

Dataset	AbsRel↓	RMSE↓	$\delta1$↑	$\delta2$↑	$\delta3$↑
DIODE	0.2509 (±1.5425e-03)	6.1306 (±7.8108e-02)	0.7197 (±8.6626e-04)	0.8563 (±3.2173e-04)	0.9183 (±3.1710e-04)
ETH3D	0.0638 (±7.9699e-04)	2.3297 (±1.0081e-01)	0.9668 (±1.5741e-04)	0.9906 (±1.1455e-04)	0.9959 (±1.2229e-04)
KITTI	0.0894 (±6.3731e-05)	4.0143 (±4.3817e-03)	0.9196 (±1.9631e-04)	0.9834 (±7.4731e-05)	0.9962 (±4.5661e-05)
NYUv2	0.0540 (±7.1042e-05)	0.2574 (±4.9001e-04)	0.9684 (±1.2029e-04)	0.9919 (±1.0664e-04)	0.9972 (±2.9471e-05)
ScanNet	0.0603 (±3.0487e-04)	0.1770 (±8.4571e-04)	0.9565 (±6.6393e-04)	0.9887 (±3.7136e-04)	0.9966 (±8.8298e-05)

Table E: **Lotus-D's results in zero-shot affine-invariant depth estimation, using true-depth space.** Here we report the additional results evaluated on DIODE dataset.

Dataset	AbsRel↓	RMSE↓	$\delta1$↑	$\delta2$↑	$\delta3$↑
DIODE	0.3258	3.9068	0.7442	0.8816	0.9341

Table F: **Lotus-D's results in zero-shot affine-invariant depth estimation, using disparity space.** Here we report the additional results evaluated on DIODE dataset.

Dataset	AbsRel↓	RMSE↓	$\delta1$↑	$\delta2$↑	$\delta3$↑
DIODE	0.2473	6.2254	0.7269	0.8579	0.9190

Table G: **Lotus-G's results in zero-shot surface normal estimation.** 10 independent runs are conducted for calculating the mean and std value. The results on OASIS is also reported.

Dataset	Mean↓	Med.↓	11.5°↑	22.5°↑	30.0°↑
iBims-1	17.4988 (±2.7417e-02)	30.7278 (±2.5334e-02)	66.0755 (±1.2659e-01)	79.0048 (±5.5158e-02)	82.6573 (±4.3533e-02)
NYUv2	16.9408 (±6.8807e-03)	26.5460 (±1.0066e-02)	59.1155 (±1.7356e-02)	77.3557 (±5.5966e-03)	83.2440 (±6.5455e-03)
OASIS	24.9383 (±1.8569e-02)	33.5827 (±4.8186e-02)	27.6380 (±7.0119e-02)	59.9048 (±4.4011e-02)	73.0071 (±5.2791e-02)
ScanNet	15.2911 (±3.0187e-02)	24.6885 (±7.0719e-02)	63.9744 (±4.0399e-02)	80.2099 (±3.6255e-02)	85.2410 (±7.3020e-02)
Sintel	35.2239 (±5.1357e-02)	45.2638 (±9.9847e-02)	19.8710 (±3.6971e-02)	42.1712 (±1.5937e-02)	54.7647 (±1.4493e-02)

Table H: **Lotus-D's results in surface normal estimation.** Here we report the additional results evaluated on OASIS dataset.

Dataset	Mean↓	Med.↓	11.5°↑	22.5°↑	30.0°↑
OASIS	25.6502	18.7791	27.5248	58.7390	71.6746

For monocular depth estimation, no matter in true-depth space (Tab. C) and disparity space (Tab. D), the variance is usually in $e^{-3} \sim e^{-5}$, which is ignorable. While in normal estimation, as illustrated in Tab. G, though the variance is lager, the mean values correspond to the Tab. 2 of main paper correctly. We also report the performance evaluated on additional benchmarks: DIODE (Tab. E, and F) and OASIS (Tab. H), for depth and normal estimation, respectively.

I Comparison between Lotus-G and Diffusion-E2E-FT

Please see Tab. I for the comparisons between Lotus-G and Diffusion-E2E-FT (Garcia et al., 2024) in monocular depth estimation.

Table I: **Quantitative comparison on zero-shot affine-invariant depth estimation** between Lotus-G and Diffusion-E2E-FT (Garcia et al., 2024). The best and second best performances are highlighted. **Lotus-G** outperforms Diffusion-E2E-FT. Here we copy the metrics from the original Diffusion-E2E-FT paper. Rankings are calculated on AbsRel↓, $\delta1$↑, and training data.

Method	Training Data	NYUv2 (Indoor)			KITTI (Outdoor)			ETH3D (Various)			ScanNet (Indoor)			DIODE (Various)			Avg. Rank
		AbsRel↓	$\delta1$↑	$\delta2$↑	AbsRel↓	$\delta1$↑	$\delta2$↑	AbsRel↓	$\delta1$↑	$\delta2$↑	AbsRel↓	$\delta1$↑	$\delta2$↑	AbsRel↓	$\delta1$↑	$\delta2$↑	
Lotus-G-Depth	59K	5.4	96.6	99.2	11.3	87.7	97.8	6.2	96.1	99.0	6.0	96.0	99.0	33.1	73.6	87.6	2.0
Diffusion-E2E-FT	74K	5.4	96.5	-	9.6	92.1	-	6.4	95.9	-	5.8	96.5	-	30.3	77.6	-	1.7
Lotus-G-Disparity	59K	5.4	96.8	99.2	8.9	92.0	98.3	6.4	96.7	99.1	6.0	95.7	98.9	25.1	71.2	85.6	1.6

J The effect of different time-steps t in one-step diffusion

In Sec. 4.2 of our main paper, we reduce the number of training time-steps of diffusion formulation to only one, and fixing the only time-step t to T. In this section, we evaluate the effect of different time-steps t in one-step diffusion, rather than exclusively fixing $t = T$. Since our model follows the diffusion formulation, which predicts the annotation starting from noise in one step, the input to the denoiser model remains the concatenation of Gaussian noise and the image latent. As shown in Tab. J, we conduct experiments on Hypersim dataset (Roberts et al., 2021) and evaluated on NYUv2 dataset (Silberman et al., 2012), without employing the detail preserver or mixture dataset training.

The results indicate that the model performs best when $t = T$ ($t = 1000$). Changing t leads to a slight degradation in performance.

Table J: **The effect of different time-steps t in one-step diffusion.** In this experiment, the models are trained on Hypersim dataset (Roberts et al., 2021) and evaluated on NYUv2 dataset (Silberman et al., 2012), without employing the detail preserver or mixture dataset training.

Time-step	$t = 1000$	$t = 750$	$t = 500$	$t = 250$	$t = 1$
AbsRel ↓	**5.587**	5.631	5.727	5.663	5.737
$\delta 1$ ↑	**96.272**	96.165	96.087	96.141	96.080

K Applications of Lotus

Thanks to its superiority, Lotus can seamlessly support a variety of applications. Fig. F illustrates four key applications: ① *Depth to Point Cloud.* The depth maps estimated by Lotus are projected into 3D point clouds; ② *Joint Estimation.* By incorporating a task switcher, Lotus can perform multiple tasks simultaneously, such as joint depth and normal map estimation with 100% shared network parameters; ③ *Single-View Reconstruction.* Using Lotus's normal predictions, high-quality meshes can be reconstructed through through Bilateral Normal Integration (Cao et al., 2022); ④ *Multi-View Reconstruction.* Leveraging per-view depth and normal predictions from Lotus, high-quality meshes can be reconstructed with MonoSDF (Yu et al., 2022), **without RGB supervision**, showcasing Lotus's robustness and accurate spatial understanding. These applications emphasize the importance of Lotus in the field of computer vision. Its accuracy and efficiency will help in addressing increasingly complex problems.

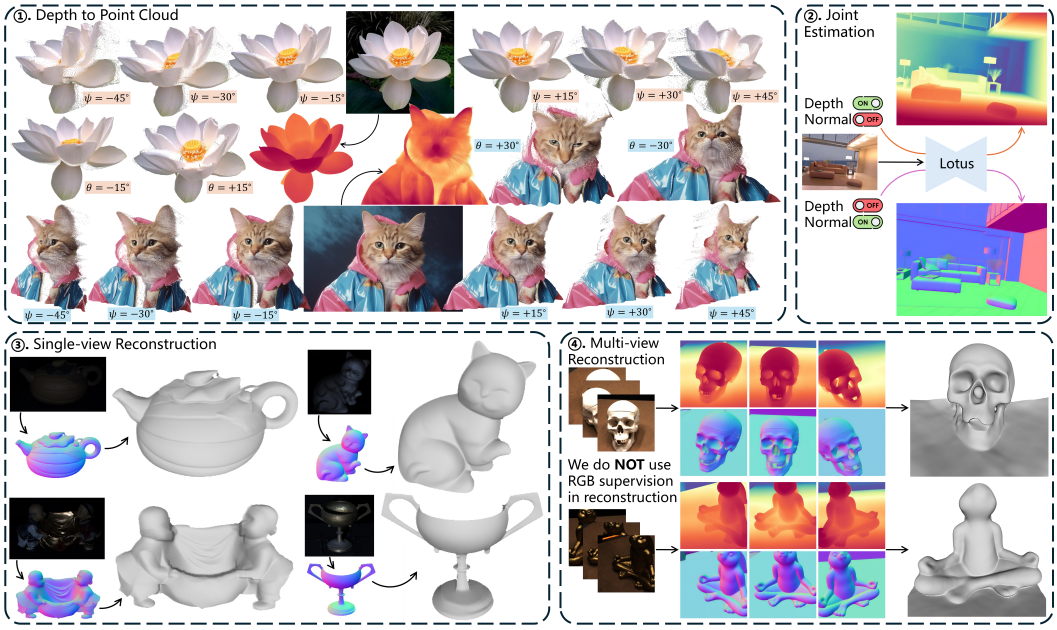


Figure F: **Applications of Lotus.** ① *Depth to 3D Point Clouds*. ② *Joint Estimation:* Simultaneous depth and normal estimation with 100% shared parameters. ③ *Single-View Reconstruction:* Reconstructing 3D meshes from normal predictions. ④ *Multi-View Reconstruction:* Reconstructing high-quality meshes using depth/normal predictions **without RGB supervision**.

L Future Work

While we have applied Lotus to two geometric dense prediction tasks, it can be seamlessly adapted to other dense prediction tasks requiring per-pixel alignment with great potential, such as panoramic

segmentation and image matting. Additionally, our performance is slightly behind DepthAnything (Yang et al., 2024a) which utilizes large-scale training data. In the future, scaling up the training data, as reveal in Fig. 7 and Tab. 3 ("Mixture Dataset") of the main paper, has great potential to further enhance Lotus's performance.

REFERENCES

Gilwon Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pp. 611–625. Springer, 2012.

Xu Cao, Hiroaki Santo, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Bilateral normal integration. In *ECCV*, 2022.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. *arXiv preprint arXiv:2409.11355*, 2024.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

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*, pp. 9492–9502, 2024.

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

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.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10912–10922, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Thomas Schops, Johannes L Schonberger, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pp. 746–760. Springer, 2012.

Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. Diffusion models trained with large data are transferable visual models. *arXiv preprint arXiv:2403.06090*, 2024.

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*, pp. 10371–10381, 2024a.

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

Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *arXiv preprint arXiv:2406.16864*, 2024.

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