Diffusion Models Monocular Depth Estimation: Overcoming Challenging Conditions

Supplementary Material

This document supplements the ECCV 2024 paper "Diffusion Models for Monocular Depth Estimation: Overcoming Challenging Conditions", providing additional implementation details and deeper insights.

1 Diffusion Distilled Data

For our image generation process, we use the diffusion model-based approach described in our main paper. Specifically, we adopt the original T2I-Adapter code [13] to generate challenging scenes from easy inputs. This model can be conditioned not only by textual prompts but also by additional information like semantic segmentation, image gradients, normals, depth maps, and more. For our purposes, we primarily leverage the network's ability to generate diverse scenarios while preserving the 3D structure from a depth map. Specifically, we provide the diffusion model with depth maps computed by an existing monocular depth network using the *easy* images. These images do not contain any particularly challenging conditions for the considered depth network. As a result, this approach produces highly accurate depth maps in simpler scenarios. Simultaneously, we incorporate textual prompts alongside these maps to specify the desired challenging attributes. However, it is worth noting that for driving scenes, we observed that the text-to-image diffusion model [13], when strongly guided by depth inputs, often struggled to visually render sufficiently extreme challenging conditions. This strong depth conditioning is necessary to ensure that the 3D structure is perfectly preserved between easy and challenging images, which is crucial for our task. However, we have found experimentally that while this approach maintains structural consistency, it sometimes limits the model's ability to generate truly challenging scene properties, especially in cases such as nighttime. To address this limitation, we implemented a two-step process that still relies entirely on diffusion models without using any real adverse images. First, we used Stable Diffusion [1] to randomly generate highly challenging images without conditioning on depth, based solely on textual prompts describing extreme conditions. Then, we passed these synthetically generated challenging images as condition to T2I-Adapter [13] along with the depth map aligned with the easy image and the desired text prompt. This approach allowed us to generate images with very low visibility, especially for nighttime scenes, resulting in truly challenging scenarios for monocular depth estimation networks, while still preserving the underlying 3D structure of the unchallenging scene. Importantly, this method maintains our commitment to using only generated challenging data, as both steps utilize diffusion models.

Moreover, before providing depth information to T2I-Adapter [1], we consistently verify the resolution of the input depth map. If it is smaller than 1 Mpx, we resize it to that resolution while keeping the same aspect ratio. We have found that this approach produces higher quality and more realistic images for the chosen diffusion model in all scenarios. After the generation process, we resize the image back to its original resolution. Regardless of whether we are generating images for ToM or driving scenarios, this procedure is applied consistently.

Given the versatility of diffusion models in creating highly diverse settings, our primary focus centers on generating scenarios that pose significant challenges for monocular depth networks. These challenges include scenarios in autonomous driving under adverse environmental conditions such as rain, night-time scenarios as well as the representation of non-Lambertian surfaces. However, it is important to note that the potential of diffusion models is not confined solely to these instances but extends to various other scenarios as well.

Regarding autonomous driving, we draw inspiration from experiments conducted by [8] on two major datasets, nuScenes [4] and RobotCar [11]. To this aim, starting from real autonomous driving scene images devoid of adverse atmospheric conditions, our approach effectively produces images portraying complex situations. This method establishes a coupling between *easy* and *challenging* images, which can be effectively leveraged in the fine-tuning phase of a monocular depth network, as described in the methodology section of the main paper.

For nuScenes [4], we use 15,129 RGB images labeled as *easy* (clear daytime conditions) to generate two distinct scenarios. Following a similar approach to [8], we create rainy and nighttime scenes, totaling 30,258 images. However, our method differs crucially from [8] in that we exclusively use simple daytime samples to generate random challenging conditions. In contrast, [8] requires access to real challenging condition images from the target domain during training, matching the test conditions (e.g., noise, luminosity, reflectivity). Our approach does not need prior knowledge or real samples of specific challenging characteristics in the target domain, which are typically unknown in real-world applications.

For this purpose, we use random textual prompts each for nighttime and rainy scenarios. Examples of these prompts, chosen arbitrarily, include:

- *"In a suburban neighborhood, rain-flooded streets reflect the warm glow of streetlamps distorted by deluge puddles overwhelming windshield wipers"*
- *"Navigating a winding mountain road, the asphalt mirrors surrounding peaks distorted by rippling puddles as windshield wipers fight the relentless downpour"*
- *"Driving at city intersections with headlight glare, dense fog, obscured vision, treacherous navigation, wet streets, rain, and roads reflecting the foggy sky"*
- ...

We want to emphasize that since text prompts are arbitrary and potentially infinite in number, conducting an exhaustive analysis of how these prompts

[1] For our experiments, we used the *TencentARC/t2i-adapter-depth-midas-sdxl-1.0*, *stabilityai/stable-diffusion-xl-base-1.0* and *madebyollin/sdxl-vae-fp16-fix* models.

might impact a downstream monocular depth network during the fine-tuning phase would be challenging and impractical, if not unfeasible. Our focus in this work is to demonstrate that even entirely random choices of the specific target environment can effectively address the problem at hand.

For the RobotCar experiments, we employ a set of prompts of the same nature as those used previously, generating a total of 17,790 challenging images from their corresponding *easy* counterparts, similarly to [8].

To create images depicting objects with non-Lambertian surfaces, we maximized the potential of diffusion models. We created approximately 20,000 images at 1024×1024 resolution using Stable Diffusion [1], focusing on various categories of challenging objects such as "bottles," "goblets," "kettles," "glasses," and more. These categories are flexible and can be expanded based on specific needs. Initially, we generated images of common objects with primarily opaque surfaces, which are more manageable for a monocular network to process. Subsequently, T2I-Adapter is employed to transform these images into their challenging counterparts featuring non-Lambertian surfaces. Below is a subset of prompts utilized to generate these *easy* objects via Stable Diffusion:

- *"Resting on a wooden closed countertop, matte wooden vessels embody the shapes of two red wooden kettles, one black wooden jar, and tow yellow wooden cups."*
- *"On a weathered wooden table, matte cardboard containers resemble various shapes of two milk cartons, one cereal boxe, and two folded paper bags. Their textured, non-reflective surfaces evoke a rustic charm, blending seamlessly into a casual dining setting."*
- *"On a wooden table, matte black wooden objects mimic pink cups, a blue teapot, and a red bowl."*
- ...

Subsequently, the conditional diffusion model [13] transforms them into non-Lambertian variations. Examples of textual prompts used for this process include:

- *"Crystal-clear, see-through glasses on a table. The beige table surface and room behind are perfectly visible through the transparent glass, as if looking through windows."*
- *"Transparent glasses on a wooden table. The beige wall and room behind are clearly visible through the glass. Mirrors on the wall reflect the scene. Each glass's outline is barely discernible, defined only by subtle light refractions."*
- *"Transparent goblets on a table. Each goblet is half-filled with different alcoholic beverages (red wine, white wine, whiskey, cognac). The table is clearly visible through the goblets. Each goblet's silhouette is barely perceptible."*
- ...

Similarly to the autonomous driving scenario, the selection of textual prompts and their variations is entirely arbitrary and potentially infinite. Our goal is to demonstrate the feasibility of this image generation process. However, in-depth investigation into each prompt is deferred to future research.

2 Additional Training Details

To ensure fair comparisons with similar methodologies, we follow the training protocols and frameworks established by [8] to handle adverse weather conditions, and [6] concerning transparent and reflective objects. Specifically, we employ the same monocular networks—utilizing md4all for [8] and DPT-Large for [6]—while seamlessly integrating their codebase into our setup. We substitute their datasets with our generated datasets derived from the diffusion model.

In their study, [8] conducted a comparative analysis between two variants of their framework, md4all-AD and md4all-DD, identifying md4all-DD as the superior performer. This conclusion directly influenced our decision to incorporate md4all-DD into our methodology.

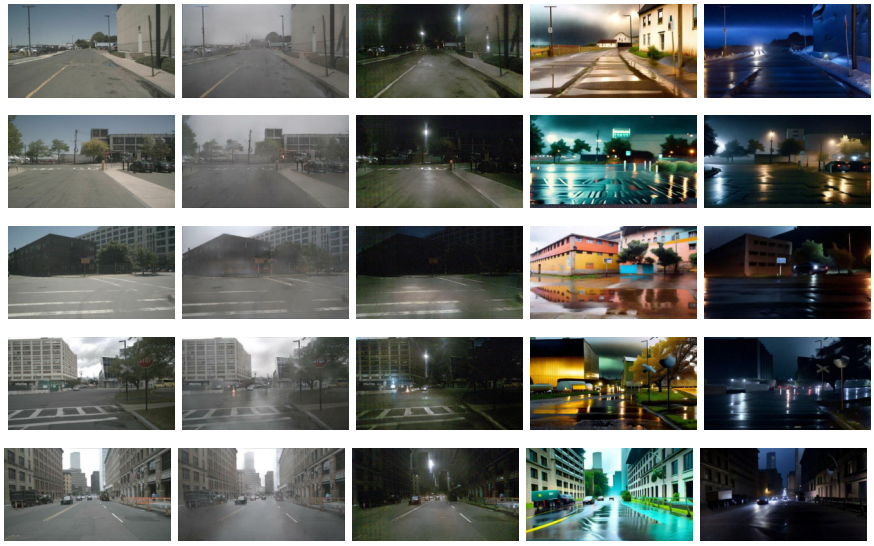
Regarding the training specifics in [8] for the md4all-DD framework, they employ knowledge distillation from a baseline depth network (B) previously trained on *easy* scenarios to a new student depth model (DD). This process enable the student model to emulate the output behavior of the teacher model even when handling challenging scenarios. During inference, the models utilize a ResNet-18 backbone and learn from image triplets sized at 576×320 for nuScenes and 544×320 for RobotCar. Operating with a single RGB input during inference ensure an equal distribution of inputs across each condition.

In [6], their approach to handling non-Lambertian scenarios involve the use of pre-trained weights from DPT-Large. Specifically, these weights are fine-tuned using images portraying non-Lambertian objects sourced from Trans10K and MSD datasets. The fine-tuning procedure extend across 20 epochs, maintaining a batch size of 8, and applying a learning rate set at 10^{-7} with exponential decay (gamma 0.95). Furthermore, they incorporate the loss function proposed in [15] to guide and optimize the training process.

For additional experiments involving other networks such as DPT-Large [16], MiDaS [15], ZoeDepth-NK [2], and Depth Anything-ViT-B [20], we incorporate our framework and internal protocol. Specifically, the self-distillation phase involves fine-tuning these networks starting from their official pre-trained weights. For DPT and MiDaS, we use a batch size of 8 and train for 30,000 iterations. Depth Anything is trained with a batch size of 8 for 5,000 iterations, while ZoeDepth uses a batch size of 3 for 30,000 iterations. In all cases, we start with an initial learning rate of 10^{-6}, later reduced to 10^{-7} after 25,000 iterations (or proportionally for Depth Anything). We adopt the same loss function proposed in [15]. To align with pre-training resolutions, image manipulations including padding, cropping, and resizing are implemented. We maintain 384 pixels for either the long or short side for most networks, except for Depth Anything, where we use 518 pixels. We preserve aspect ratios through square cropping. The AdamW optimizer [10] is employed for training. Augmentations, including horizontal flips, color jittering, RGB shift, Gaussian noise, defocusing, and motion blur [3], are applied consistently across all experiments.

3 Qualitative Comparison – Generated Images

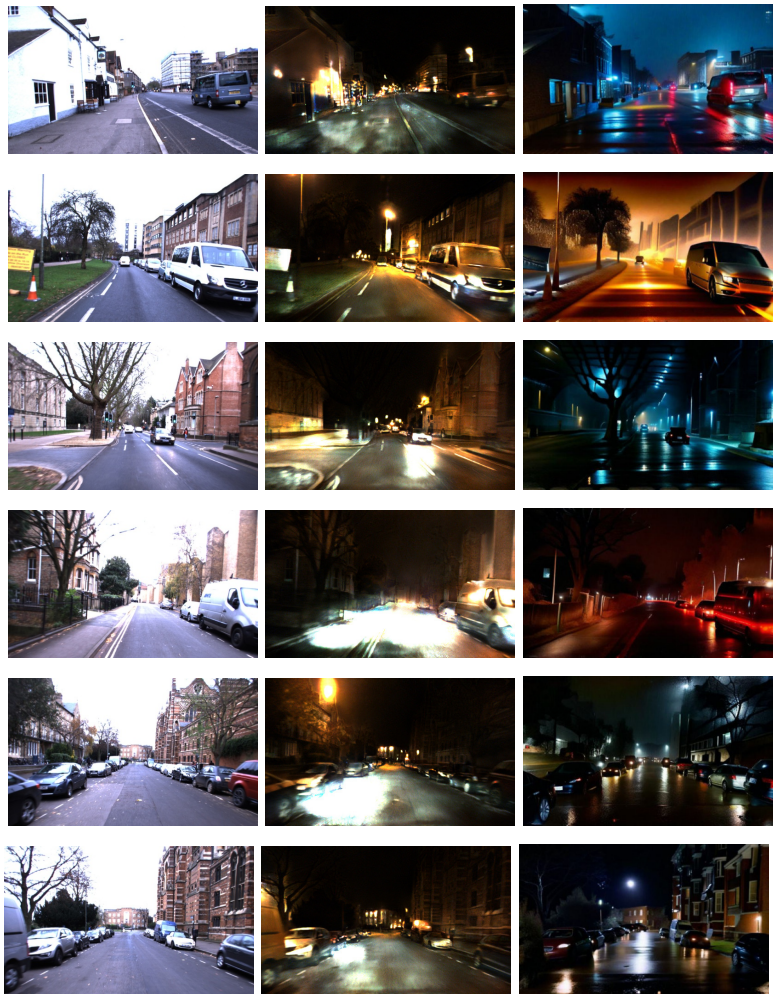
We collect some examples of images generated by [8] with ForkGAN [22], and our pipeline using T2I-Adapter [13]. **NuScenes [4].** Figure I reports examples from *nuScenes* [4]. In the leftmost column (a), we present real images from the *day-clear* domain. Columns (b) and (c) show ForkGAN-generated day-rain and night samples, while (d) and (e) display our pipeline's outputs for the same domains. The pipeline in [8] mainly replicates rain artifacts and nighttime sensor noise, due to using real images. Conversely, our images are more diverse, focusing on light glare and road reflections without relying on images of real, challenging conditions in the target domain.



(a) *day-clear* **(b)** → *day-rain* [8] **(c)** → *night* [8] **(d)** → *day-rain* [13] **(e)** → *night* [13]

Fig. I: From left to right: the original *day-clear* RGB image sourced from nuScenes [4]; the *day-rain* and the *night* images generated by [8] employing ForkGAN [22]; the *day-rain* and the *night* images produced by our method employing T2I-Adapter [13].

RobotCar [11]. In analogy, Figure II shows real images from RobotCar, *day* domain. These are followed by nighttime images generated with ForkGAN [22] (b) and our method based on T2I-Adapter (c). Again, the former method reproduces some of the properties intrinsic to the RobotCar dataset – such as the strong motion blur observed in most real nighttime images from the dataset itself. In contrast, the images we generated with our pipeline are more general and not strictly bound to the effects observed from RobotCar, since our method does not exploit any nighttime real sample from it.



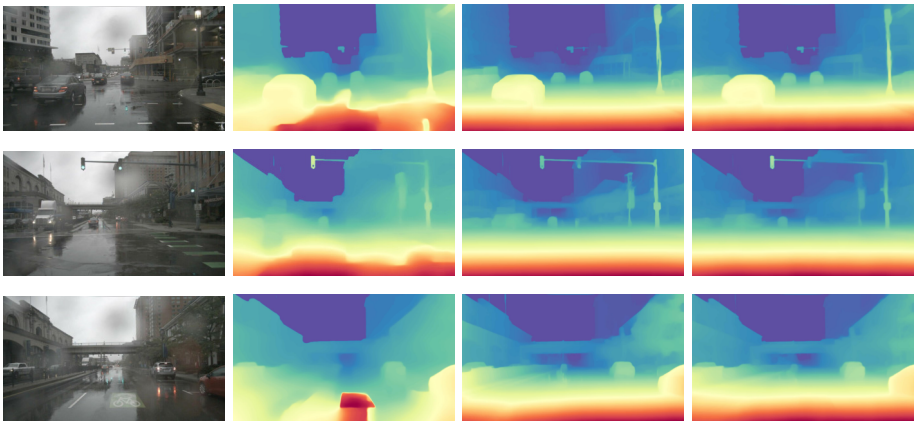
(a) *day* **(b)** → *night* [7] **(c)** → *night* [13]

Fig. II: From left to right: the original *day* RGB image sourced from the RobotCar dataset [11], followed by the *night* image generated by [8] employing ForkGAN [22]; the *night* image generated by our method using T2I-Adapter [12].

4 Qualitative Comparison – Predicted Depth Maps in Driving Scenarios

We now conduct a comparative analysis of the depth maps predicted by baseline models [16], the approach proposed by [8], and our method across various challenging driving scenarios.

NuScenes [4] – *day-rain.* Figure III depicts three samples extracted from the nuScenes dataset, specifically from the *day-rain* domain (a). DPT model's predictions [16] (b) cannot infer the correct depth due to the challenging rain-induced reflections on the road surface. Conversely, fine-tuning this model with data either generated by [8] (c) or by our approach (d) remarkably enhances the accuracy of predicting the road plane. Additionally, we can appreciate finer detail in the predicted depths, such as distant cars, previously absent in the original predictions.

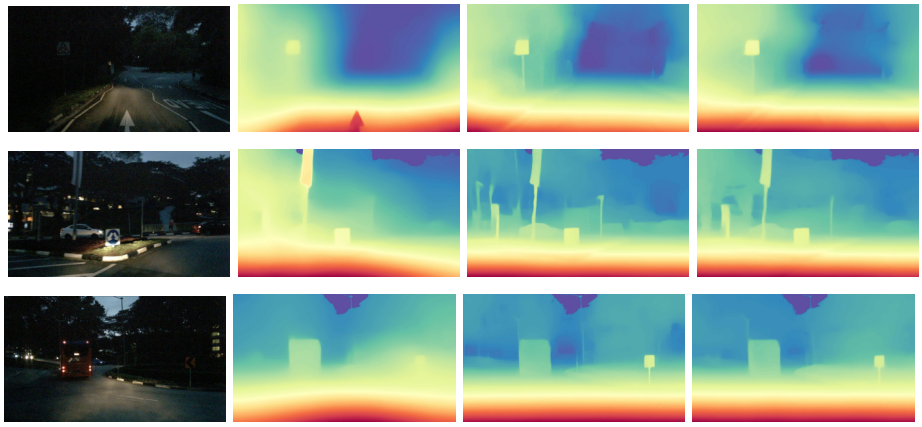


(a) RGB - *day-rain* **(b)** DPT [16] – Original **(c)** DPT [16] ft. Gasperini et al. [8] **(d)** DPT [16] ft. Ours

Fig. III: From left to right: RGB image sourced from the nuScenes [4] validation set (*day-rain*), the corresponding depth map generated by the original DPT [16], followed by the fine-tuned DPT using the method proposed in [7]. Finally, DPT fine-tuned employing our proposed diffusion-based method.

NuScenes [4] – *night.* The comparison in Figure IV focuses on NuScenes within the *night* domain, showing three specific examples (column (a)). Again, the original DPT predicts depth maps lacking several details, often failing in the darkest areas of the image, e.g., where car lights do not brighten the scene. The fine-tuning carried out on generated images (b,c) allows for recovering most of the details lost in the original predictions, even those very hard to catch by the human eye – *e.g.*, the traffic signal in the third example.

RobotCar [11] – md4all baseline [8]. In Figure V, our comparison shifts to the RobotCar dataset, where we've selected seven samples from the *night* domain (a). Here, we illustrate the shortcomings of the md4all baseline model [8].

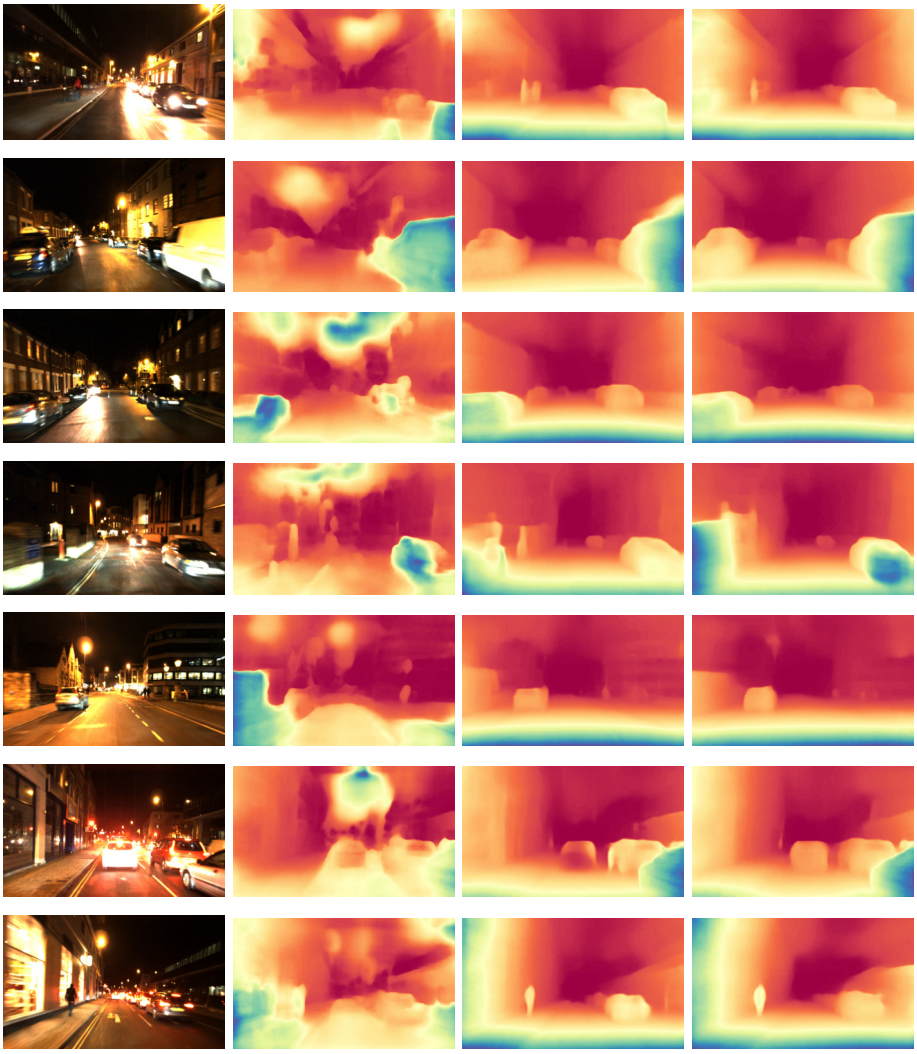


(a) RGB - *night* **(b)** DPT [16] – Original **(c)** DPT [16] ft. [8] **(d)** DPT [16] ft. Ours

Fig. IV: From left to right: RGB image sourced from the nuScenes [4] validation set (*night*), the corresponding depth map generated by the original DPT [16], followed by the fine-tuned DPT using the method proposed in [7]. Finally, DPT fine-tuned employing our proposed diffusion-based method.

This model was trained only on daytime images, so it struggles with nighttime images within the RobotCar dataset.

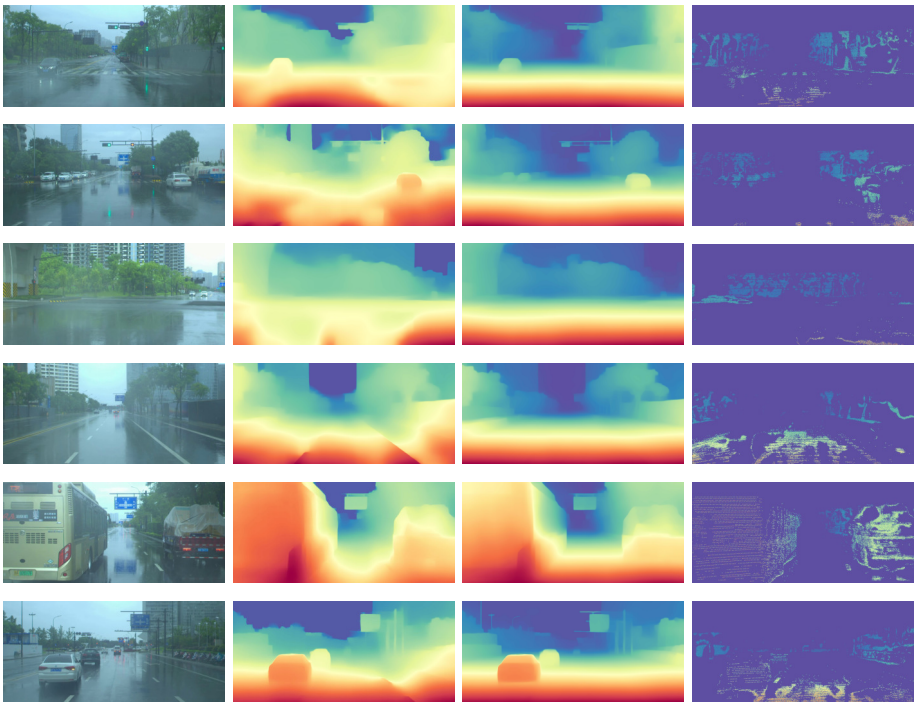
In contrast, the md4all-DD variants exhibit significantly improved robustness, particularly when trained using methods outlined in [8] (c) or our approach (d). Our model showcases enhanced performance, surpassing even the original md4all-DD instance. For instance, in the sixth row, while the original md4all-DD model partially misses the car in front of the camera. Conversely, our model, trained using our pipeline, accurately detects this car in the predicted depth map.



(a) RGB - *night* **(b)** md4all (baseline) [8] **(c)** md4all-DD [8] **(d)** md4all-DD [8] ft. Ours

Fig. V: From left to right: RGB image sourced from the RobotCar [11] test set (*night*), the corresponding depth map generated by md4all (baseline) [8], followed by the original md4all-DD proposed in [7] and md4all-DD trained using our approach.

DrivingStereo [19]. We conclude with some qualitative results from the DrivingStereo dataset, for which we report eight examples of rainy images (a) in Fig. VI. We recall the fact that other methodologies, such as [8], cannot be applied in this scenario due to the absence of both the *easy* and *challenging* images. By looking at predictions by the original DPT model [16] (b), we can notice how the wet road surface again makes it struggle to predict smooth and planar surfaces. On the contrary, by fine-tuning it on our data (c) the network learns to properly deal with such challenging regions and consistently predict more accurate depth maps. We argue that this evident difference in qualitative terms is not fully reflected in the numbers reported in the main paper. This is caused by the very sparse ground truth provided by DrivingStereo itself (d), often missing in those challenging regions where DPT originally fails, greatly disadvantaging our method, which can predict them correctly. Nevertheless, the quantitative gain remains neat in reported tables of the main paper.



(a) RGB - *day-rain* **(b)** DPT [16] **(c)** DPT [16] ft. Ours **(d)** Sparse Ground Truth

Fig. VI: From left to right: RGB image sourced from the DrivingStereo [4] dataset (*rain*), the corresponding depth map generated by the original DPT [16], followed by the fine-tuned DPT using our proposed diffusion-based method. Finally, the sparse ground truth depth map. We note that depth ground truth provides only a few valid pixels for challenging regions.

(a) RGB **(b)** Depth **(c)** "*Hazardous Smog Blanketing City Streets*" **(d)** "*Low Visibility in Urban Fog*"

(e) "*Stranded in a Car During Intense Hailstorm*" **(f)** "*Urban Roads with Extreme Sun Glare*" **(g)** "*Tropical Cyclone's Downpour* " **(h)** "*Heavy Snowfall on a Remote Forest Road*"

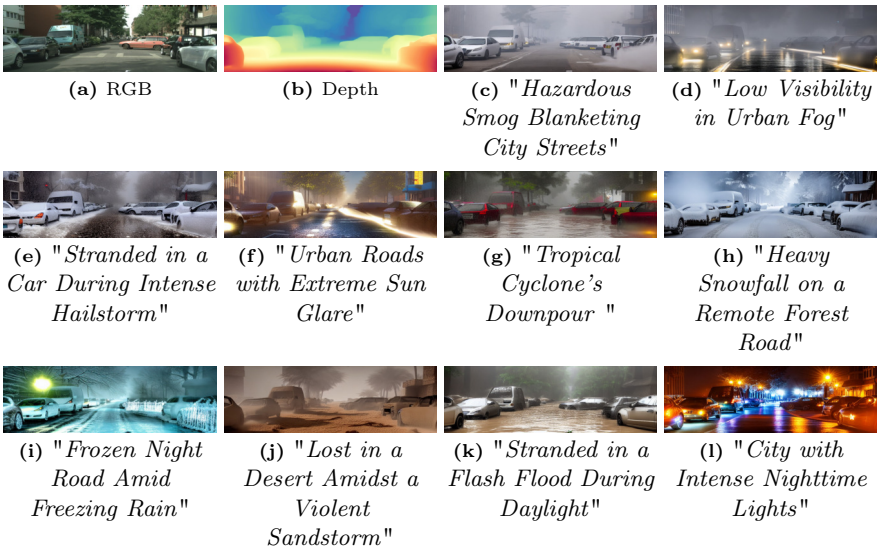
(i) "*Frozen Night Road Amid Freezing Rain*" **(j)** "*Lost in a Desert Amidst a Violent Sandstorm*" **(k)** "*Stranded in a Flash Flood During Daylight*" **(l)** "*City with Intense Nighttime Lights*"

Fig. VII: Visual examples of images generated with [13] showing different weather conditions. These transformations result from modifying text prompts within the Apolloscapes [9] dataset.

5 Qualitative Results – Playing with Text Prompts

In this section, we collect additional qualitative examples to point out the full potential of our method to generate vast amounts of challenging images over which a depth estimation network can be fine-tuned to improve its robustness.

Starting from Real Images. We provide examples of images generated starting from as few as three real images (a), respectively, from several datasets: Apolloscapes [9] in Fig. VII, CityScapes [5] in Fig. VIII and Mapillary [14] in Fig. IX. By predicting a depth map for each of these samples (b), we can generate countless images in various weather conditions by simply playing with the text prompt. Once again, this reinforces the immense potential of our technique in generating substantial volumes of training data. It holds promise for handling diverse and challenging conditions that often pose difficulties for depth estimation models. This approach simply necessitates having a set of images without such challenges as a starting point.



(a) RGB **(b)** Depth **(c)** "*Hazardous Smog Blanketing City Streets*" **(d)** "*Low Visibility in Urban Fog*"

(e) "*Stranded in a Car During Intense Hailstorm*" **(f)** "*Urban Roads with Extreme Sun Glare*" **(g)** "*Tropical Cyclone's Downpour* " **(h)** "*Heavy Snowfall on a Remote Forest Road*"

(i) "*Frozen Night Road Amid Freezing Rain*" **(j)** "*Lost in a Desert Amidst a Violent Sandstorm*" **(k)** "*Stranded in a Flash Flood During Daylight*" **(l)** "*City with Intense Nighttime Lights*"

Fig. VIII: Visual examples of generated images depicting different weather conditions. These transformations result from modifying text prompts within the CityScapes [5] dataset.

(a) RGB **(b)** Depth **(c)** "*Hazardous Smog Blanketing City Streets*" **(d)** "*Low Visibility in Urban Fog*"

(e) "*Stranded in a Car During Intense Hailstorm*" **(f)** "*Urban Roads with Extreme Sun Glare*" **(g)** "*Tropical Cyclone's Downpour* " **(h)** "*Heavy Snowfall on a Remote Forest Road*"

(i) "*Frozen Night Road Amid Freezing Rain*" **(j)** "*Lost in a Desert Amidst a Violent Sandstorm*" **(k)** "*Stranded in a Flash Flood During Daylight*" **(l)** "*City with Intense Nighttime Lights*"

Fig. IX: Visual examples of generated images depicting different weather conditions. These transformations result from modifying text prompts within the Mapillary [14] dataset.

Starting from Textual Prompts only. Our pipeline can generate an extensive array of images for fine-tuning depth estimation models, even without real images, by utilizing various text prompt combinations. Figure X illustrates this process. The top row shows five *easy* images created using Stable Diffusion [1] with text inputs only. The second row shows the depth maps predicted by Depth Anything [20] using the authors' original weights. These weights were then used with [13] to produce diverse new images featuring glasses and transparent objects. In Figure XI, we present additional examples involving various objects such as goblets, mirrors, vessels, and more.

Fig. X: Top row shows RGB images obtained from Stable Diffusion [1], portraying mostly Lambertian surfaces on simple objects. The second row displays depth maps computed by Depth Anything [20] from the simple object images in the first row. From the third row onwards, all images are generated by T2I-Adapter [13] to transform the simple object images into visuals featuring non-Lambertian surfaces, thereby incorporating reflective and transparent elements.

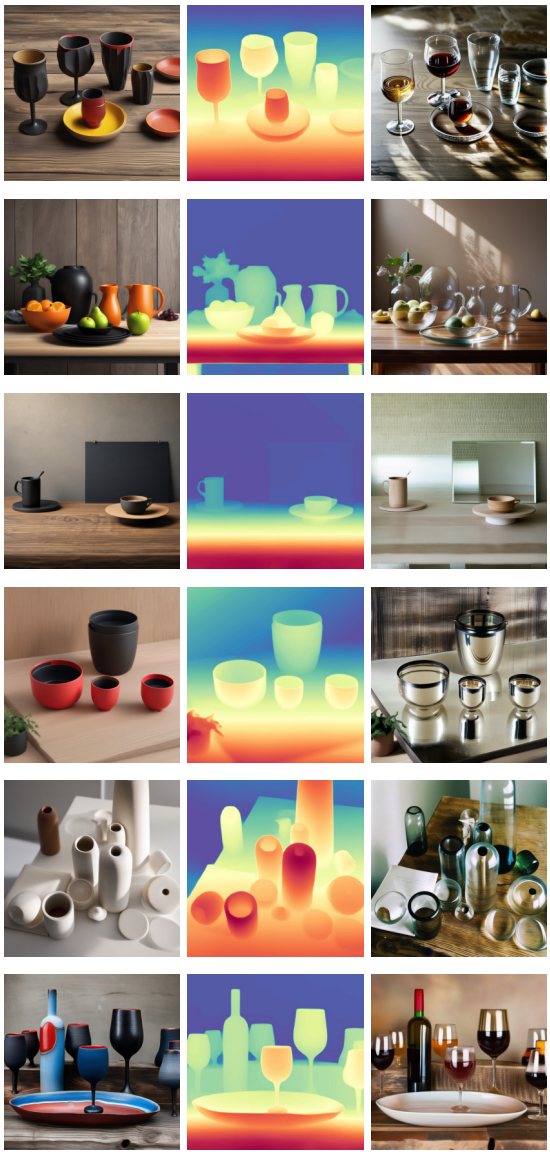
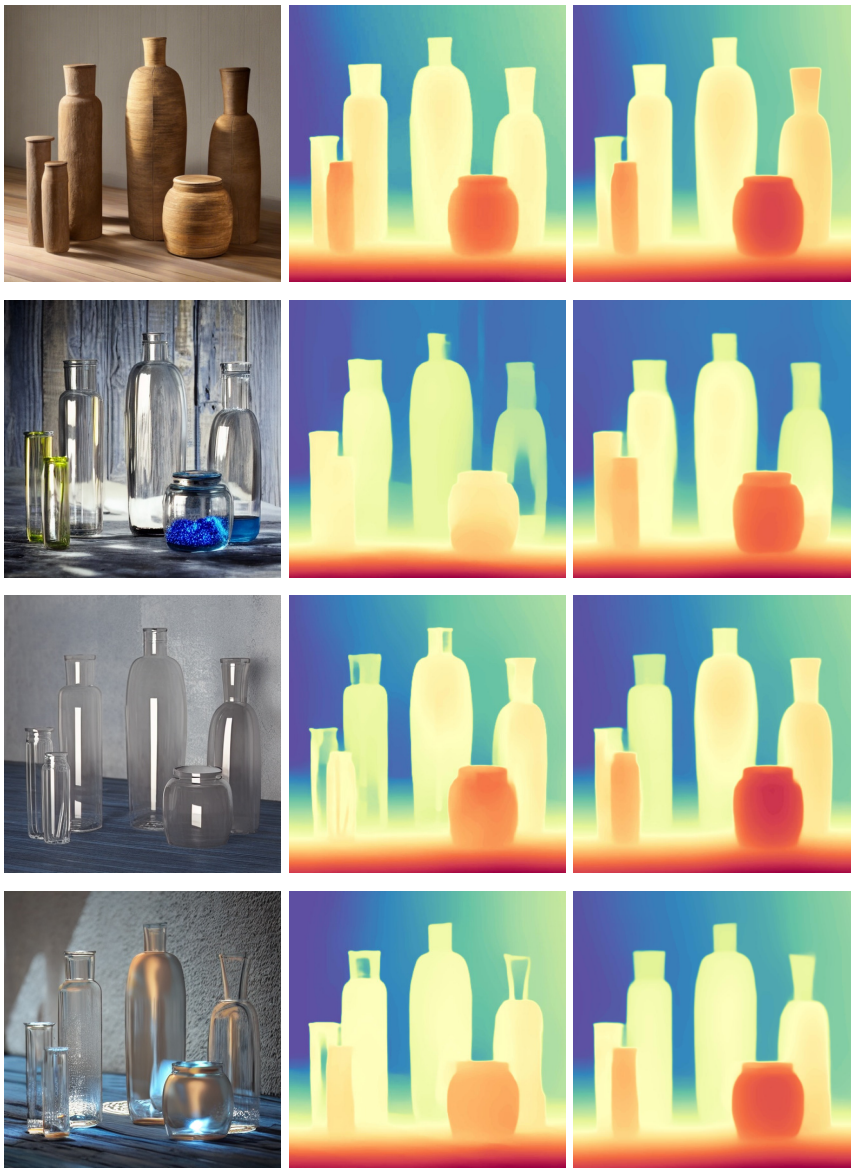


Fig. XI: On the left, RGB images computed by Stable Diffusion [1], showing mostly Lambertian surfaces on simple objects. The second row shows depth maps computed by Depth Anything [20] from the simple object images in the left column. On the right, all images are generated by T2I-Adapter [13] to transform the simple object images into visuals featuring non-Lambertian surfaces, thus including reflective and transparent elements.

6 Qualitative Results – Depth Maps on Transparent Objects

Finally, we report some examples of predicted depth maps for images framing transparent objects.

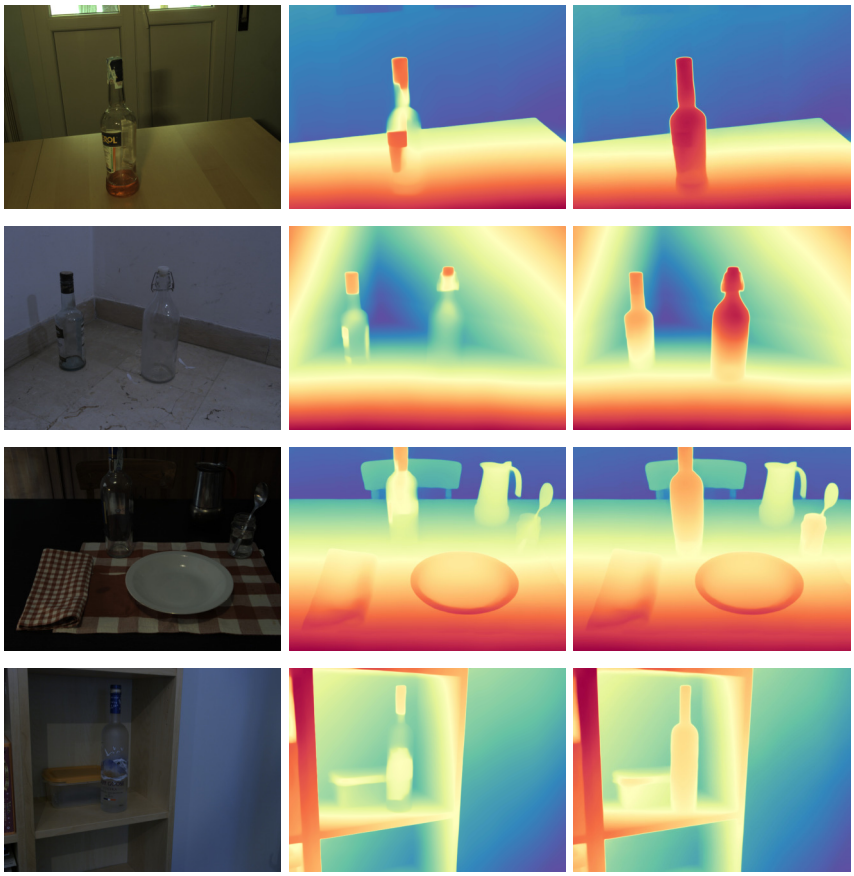
Generated data. The depth estimation models are challenged by the images generated by our pipeline. As proof of this, Fig. XII shows four images on the leftmost column (a), followed by depth maps predicted by DPT before (b) and after (c) being fine-tuned on our data. On the first row, we show an *easy* image framing wooden objects, generated through Stable Diffusion [1] with text prompts solely. On the remaining rows, we show three images obtained by running [13] to generate challenging samples. On these samples, we can notice how the depth maps predicted by the original DPT [16] model (b) are inaccurate on some of the transparent objects, while they appear much more accurate after our fine-tuning (c).



(a) RGB **(b)** DPT [16] – Original **(c)** DPT [16] – Fine-Tuned

Fig. XII: The uppermost row within the initial column features the original *easy* image generated by Stable Diffusion [1], illustrating simple objects. Subsequent rows in the same column exhibit T2I-generated [13] images simulating scenarios involving non-transparent, and reflective surfaces. Transitioning to the second column reveals the corresponding depth maps predicted by the DPT [16] using the original weights provided by the authors. Lastly, the third column presents depth maps derived from the same network, refined through the implementation of our methodology.

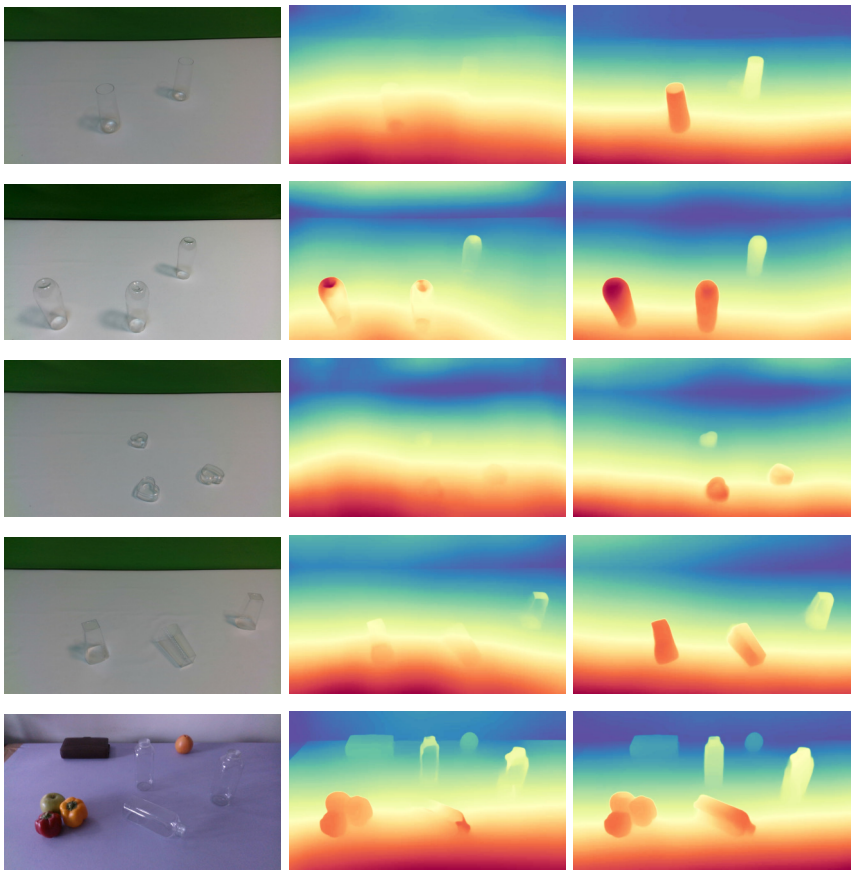
Booster dataset [21]. Fig. XIII depicts four images from the Booster training set (a), used in our evaluation. In the middle, we report the depth maps predicted by Depth Anything [20] using the original weights made available by the authors (b). Despite being accurate in most regions of the images, sometimes it fails to properly estimate the depth of some transparent parts in the objects – *e.g.*, as in the bottle on the top row. However, after fine-tuning it on our data, Depth Anything can correctly recover the real 3D structure of any object in the scene (c).



(a) RGB **(b)** Depth Anything [20] – Original **(c)** Depth Anything [20] – Fine-Tuned

Fig. XIII: From left to right: RGB image sourced from the Booster [21] training set, corresponding depth map generated by the original Depth Anything [20], and fine-tuned Depth Anything utilizing our method.

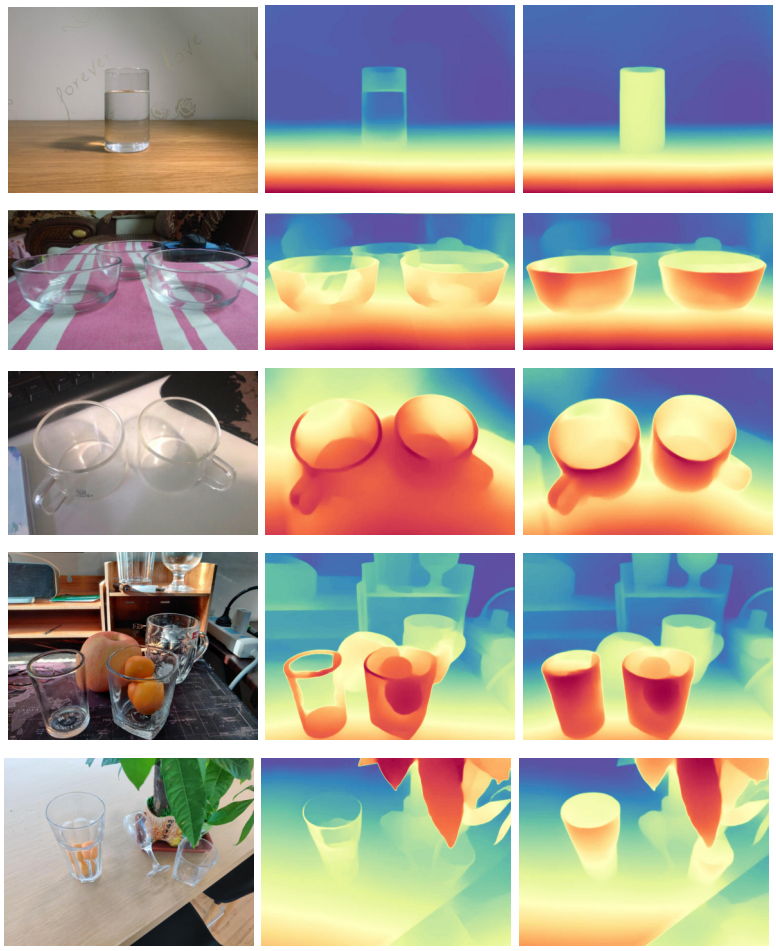
ClearGrasp dataset [17]. Fig. XIV shows five particularly challenging examples (a) from the ClearGrasp dataset. In the middle column, we can see how DPT [16] initially fails to recognize most of the transparent objects present in the scenes (b). Nonetheless, after being fine-tuned on our data generated from text prompts solely, it can properly predict consistent depth for any of them (c).



(a) RGB **(b)** DPT [16] – Original **(c)** DPT [16] – Fine-Tuned

Fig. XIV: From left to right: RGB image sourced from ClearGrasp [17], corresponding depth map generated by the original DPT [16], and fine-tuned DPT utilizing our method.

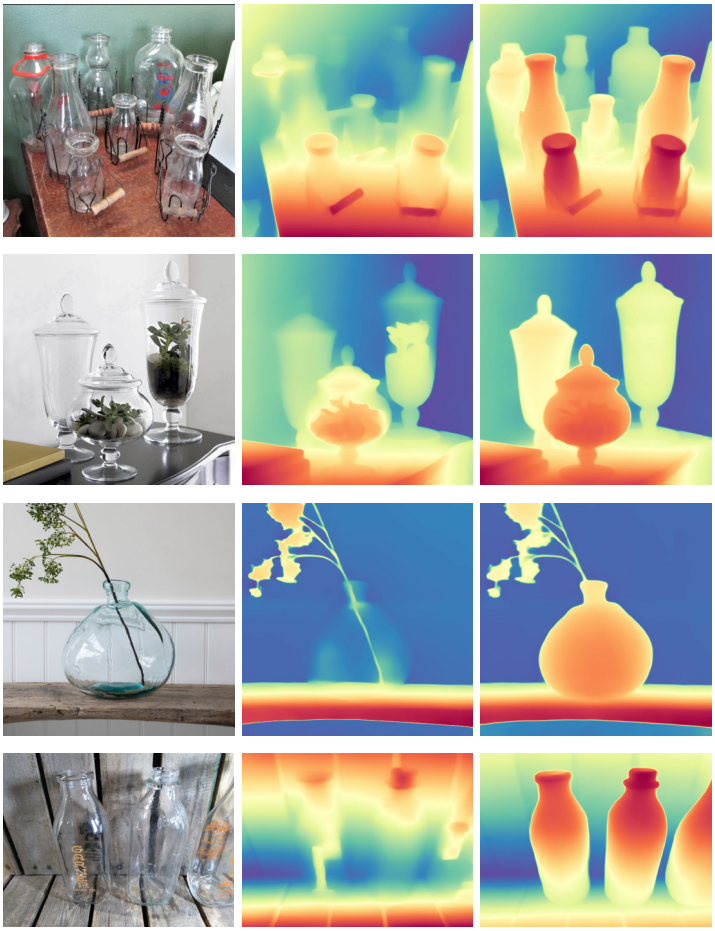
Trans10K dataset [18]. We collect some additional, qualitative results on the Trans10K dataset, which unfortunately does not provide ground-truth depth for quantitative evaluations. Fig. XV shows six examples from this dataset (a), followed by the depth maps predicted by DPT [16] before (b) and after (c) fine-tuning it on our data. Again, we can appreciate how our strategy allows for greatly improving the perception of transparent objects.



(a) RGB **(b)** DPT [16] – Original **(c)** DPT [16] – Fine-Tuned

Fig. XV: From left to right: RGB image sourced from Trans10K [18], corresponding depth map generated by the original DPT [16], and fine-tuned DPT utilizing our method.

Random Pictures from the Web. In conclusion, Fig. XVI shows web-sourced images (a), with depth maps from Depth Anything [20] before (b) and after (c) fine-tuning using our data.



(a) RGB **(b)** Depth Anything [20] – Original **(c)** Depth Anything [20] – Fine-Tuned

Fig. XVI: Left to right: Web-sourced RGB image, depth map from original Depth Anything [20], and depth map from our fine-tuned Depth Anything.

7 Limitations

The application of our method comes with certain limitations. Firstly, the reliance on pre-trained diffusion models that require a substantial amount of data for training. The quality and diversity of the generated images are dependent on the training data and the architecture of the diffusion models. Additionally, the effectiveness of the chosen T2I-Adapter [13] model is contingent on the source of the training data, predominantly derived from models like MiDaS [15], potentially introducing biases. While the 3D structure is generally well-preserved between *easy* and *challenging* images, there can occasionally be discrepancies that might lead to slightly different 3D structures, potentially impacting the accuracy and reliability of the depth estimation results. Finally, despite the flexibility offered by text prompts in shaping scenarios, achieving precise control over the outputs of diffusion models remains a challenge. The text prompts may not always capture the desired characteristics or attributes of the generated scenes, highlighting the need for further research and development of more advanced control mechanisms to achieve finer-grained control over the generated images.

References

1. Stable diffusion xl - sdxl 1.0 model card (2023), https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0 1, 3, 13, 14, 15, 16, 17
2. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) 4
3. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: Fast and flexible image augmentations. Information **11**(2) (2020). https://doi.org/10.3390/info11020125, https://www.mdpi.com/2078-2489/11/2/125 4
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) 2, 5, 7, 8, 10
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 11, 12
6. Costanzino, A., Zama Ramirez, P., Poggi, M., Tosi, F., Mattoccia, S., Di Stefano, L.: Learning depth estimation for transparent and mirror surfaces. In: The IEEE International Conference on Computer Vision (2023), iCCV 4
7. Gasperini, S., Koch, P., Dallabetta, V., Navab, N., Busam, B., Tombari, F.: R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In: 2021 International Conference on 3D Vision (3DV). pp. 751–760. IEEE (2021) 6, 7, 8, 9
8. Gasperini, S., Morbitzer, N., Jung, H., Navab, N., Tombari, F.: Robust monocular depth estimation under challenging conditions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023) 2, 3, 4, 5, 6, 7, 8, 9, 10
9. Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The apolloscape open dataset for autonomous driving and its application. IEEE transactions on pattern analysis and machine intelligence **42**(10), 2702–2719 (2019) 11

10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 4
11. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research **36**(1), 3–15 (2017) 2, 6, 7, 9
12. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4296–4304 (2024) 6
13. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023) 1, 3, 5, 6, 11, 13, 14, 15, 16, 17, 22
14. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: Proceedings of the IEEE international conference on computer vision. pp. 4990–4999 (2017) 11, 13
15. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ICCV (2021) 4, 22
16. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(3) (2022) 4, 7, 8, 10, 16, 17, 19, 20
17. Sajjan, S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., Song, S.: Clear grasp: 3d shape estimation of transparent objects for manipulation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 3634–3642. IEEE (2020) 19
18. Xie, E., Wang, W., Wang, W., Ding, M., Shen, C., Luo, P.: Segmenting transparent objects in the wild. arXiv preprint arXiv:2003.13948 (2020) 20
19. Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B.: Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 899–908 (2019) 10
20. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: 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 (2024) 4, 13, 14, 15, 18, 21
21. Zama Ramirez, P., Tosi, F., Poggi, M., Salti, S., Di Stefano, L., Mattoccia, S.: Open challenges in deep stereo: the booster dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2022), cVPR 18
22. Zheng, Z., Wu, Y., Han, X., Shi, J.: Forkgan: Seeing into the rainy night. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 155–170. Springer (2020) 5, 6