DegBins: Degradation-Driven Binning for Depth Super-Resolution

Zhiqiang Yan[1] **Zhengxue Wang**[2] **Jian Yang**[2] **Gim Hee Lee**[1]
[1]Department of Computer Science, National University of Singapore
[2]Nanjing University of Science and Technology, {yanzq, gimhee.lee}@nus.edu.sg

Abstract

Depth super-resolution (DSR) aims to recover a high-resolution (HR) depth map from its low-resolution (LR) counterpart. With color image guidance, this task is typically formulated as learning the residual between HR and LR in a low-dimensional feature space. However, this additive formulation is insufficient to accurately capture the complex relationship between HR and LR, especially under spatially varying degradations. In this paper, we introduce **DegBins**, a novel DSR framework that leverages degradation-driven binning to adaptively enhance residual modeling. Specifically, DegBins reformulates the regression-based DSR as a hybrid classification-regression problem, where the residual depth is represented as a linear combination of discrete depth bins weighted by their learned probability distribution, yielding more flexible and expressive representations. Furthermore, DegBins models the degradation relationship between HR and LR in a high-dimensional feature space, enabling adaptive bin range adjustment and probability optimization conditioned on local degradation characteristics. To progressively improve reconstruction quality, DegBins adopts a multi-stage refinement scheme, where each stage performs finer-grained bin partitioning and probability updating based on the former estimation. This coarse-to-fine design facilitates more accurate depth recovery, particularly in regions with severe degradations or complex structural variations. Extensive experiments across five benchmarks demonstrate that DegBins consistently outperforms existing state-of-the-art methods in terms of accuracy, robustness, and generalization.

1 Introduction

Depth super-resolution (DSR) is a fundamental task in geometric computer vision, with broad applications in 3D reconstruction [9, 41, 3, 59, 35], augmented reality [29, 33, 50, 58], and robotics [30, 55, 22], where precise high-quality depth perception is critical for reliable scene understanding and downstream geometry-aware decision making. With guidance from color images, this task is commonly formulated as residual learning [60, 9, 13] between the target high-resolution (HR) depth and the observed low-resolution (LR) depth. However, the degradation relationship [39, 44] between HR and LR is inherently complex, especially in real-world scenarios where spatially varying blur, noise, sensor imperfections, and misalignment artifacts are prevalent, rendering the naive additive residual formulation insufficient to accurately characterize such a relationship.

To address these limitations, we propose DegBins, which leverages a degradation-driven binning strategy to guide the residual modeling from a distributional perspective instead of a deterministic regression paradigm. First, DegBins reformulates the pixel-wise regression-based DSR into a hybrid classification-regression problem. As illustrated in Fig. 1, the numerical range of residual depth is discretized into multiple depth intervals termed bins. The model then predicts a probability distribution over these bins, and computes the refined residual depth as a weighted linear combination

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

of the bin centers using the predicted probabilities. Compared with conventional DSR approaches, this design enables more flexible and expressive representations of the residual depth. Second, in contrast to learning the residual relationship in a limited low-dimensional space, DegBins exploits degradation representations between HR and LR in a high-dimensional space to enable adaptive adjustment of bin ranges and optimization of probability distributions conditioned on local degradation characteristics. This facilitates a more accurate characterization of the relationship by leveraging high-dimensional spatial dependencies to guide those in the low-dimensional space. Third, DegBins employs a multi-stage refinement scheme, in which each stage adopts finer-grained bin partitioning and updates the probability distribution based on the former prediction. This coarse-to-fine strategy progressively enhances depth quality and is particularly effective in regions with severe degradation or complex structural variations.

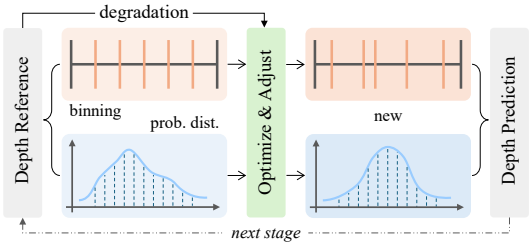


Figure 1: Conceptual core idea of our DegBins.

In summary, our main contributions are as follows:

- We present a degradation-driven binning based framework termed DegBins, which leverages high-dimensional degradation representations to enable adaptive bin range adjustment and probability distribution optimization.
- We propose a multi-stage refinement scheme through coarse-to-fine bin partitioning and probability updating based on the former estimation for high-quality depth recovery, especially in heavily degraded or structurally complex regions.
- Comprehensive evaluations demonstrate that DegBins achieves superior performances over current leading methods with notable gains in accuracy while maintaining strong robustness and generalization across diverse scenarios.

2 Related Work

Depth Super-Resolution. Early approaches mainly comprise optimization-based [7] and filtering-based paradigms [20]. They typically rely on hand-crafted priors to restore HR depth, making it difficult to accurately recover geometric boundaries in structurally complex regions. To overcome these limitations, numerous deep learning-based methods [56, 40, 12, 41] are proposed that focus on exploring cross-modal feature fusion between RGB and depth. Such methods [58, 38, 30, 31, 33] substantially enhance the LR depth by adaptively incorporating geometry-related structural cues from RGB while suppressing irrelevant texture copying. For example, Zhao et al. [55] introduce discrete cosine transform to transfer modality-shared features from RGB to depth. SGNet [37] presents a structure-guided framework to aggregate depth-consistent high-frequency components across spatial, gradient, and frequency domains. Additionally, Zheng et al. [57] design a geometry-decoupled network to disentangle fine-grained and global geometric knowledge. Zhong et al. [61] develop a cross-modality neural architecture search framework to automatically design efficient fusion models.

Degradation Modeling. Recent progress in color image restoration has been largely driven by more expressive degradation modeling strategies [49, 53, 36]. For example, [53] design a realistic and practically deployable degradation pipeline to mimic complex real-world corruptions in blind super-resolution. In a similar vein, [62] introduce a degradation-adaptive regression framework that predicts region-specific degradation for local patches. [19] propose a model in which network parameters are dynamically modulated according to the estimated degradation characteristics. Beyond single-degradation settings, a number of approaches [51, 15, 52] aim to handle multiple degradation types within a unified framework. For example, Zhang et al. [51] present a multi-degradation restoration network that progressively captures degradation patterns through clustering, eliminating the need for explicit degradation priors. Similarly, Zhang et al. [52] formulate an ingredient-oriented representation to explicitly characterize inter-image degradation relationships. More recently, Yao et al. [48] propose a neural framework that captures latent degradation factors and disentangles mixed degradations for improved restoration. DORNet [39] adaptively addresses unknown degradations in real-world scenes using implicit degradation representations.

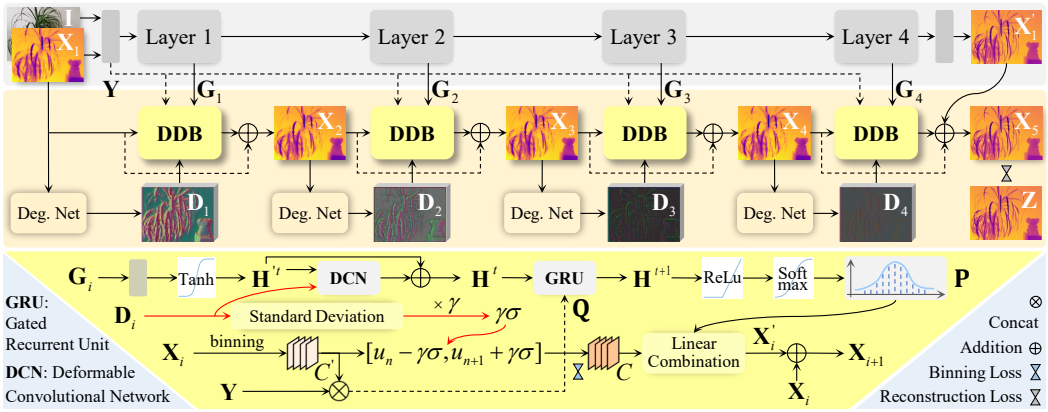


Figure 2: Framework of DegBins, where DDB represents the degradation-driven binning strategy.

Bin-Based Depth Learning. Monocular depth estimation is typically formulated as a regression problem [5, 25, 34], where pixel-wise depth values are directly predicted under a regression loss. DORN [8] first reformulates this task as a per-pixel classification problem by discretizing the continuous depth range into a set of intervals (bins). Based on this idea, AdaBins [1] adopts a hybrid classification–regression paradigm to predict the probability distribution over globally shared bins and compute the final depth via a linear combination. BinsFormer [18] further introduces a sufficient interaction between probability distribution and bin predictions, and LocalBins [2] replaces globally shared bins with pixel-wise adaptive bins. Moreover, IEBins [26] introduces elastic bins which progressively refine the bin ranges in a multi-stage manner. During the process of high-resolution depth reconstruction, all these bin-based depth estimation approaches offer valuable insights, particularly when degradation representations are available.

3 Methodology

3.1 Overview

Inspired by prior works [8, 1], we reformulate the pixel-wise regression-based DSR as a hybrid classification–regression paradigm, enabling more flexible and expressive representations that improve the model's generalization capability. Fig. 2 illustrates the pipeline of DegBins, which primarily comprises two components: the gray part representing the basic feature encoding layers and the orange part corresponding to the progressive depth prediction blocks. In the gray part, we employ the pre-trained DuCos [46] as the backbone, which takes a color image $\mathbf{I}$ and a LR depth map $\mathbf{X}_1$ as input, and outputs a fused context feature $\mathbf{Y}$, layer-specific features $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4\}$, and an initial depth estimate $\mathbf{X}_1^{'}$. In the orange part, the LR depth $\mathbf{X}_1$ is first discretized into multiple bins [1, 2, 26]. The network then infers high-dimensional degradation representations $\{\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4\}$ via pre-trained degradation blocks [39]. Conditioned on these degradation representations, the proposed degradation-driven binning (DDB) strategy adaptively adjusts the bin ranges and their associated probability distributions, yielding finer-grained depth outputs. This process is iteratively performed over four stages, where the depth output at each stage serves as the input to the next, ultimately producing the final depth prediction $\mathbf{X}_5$. Subsequently, Sec. 3.2 introduces the DDB strategy together with its multi-stage refinement scheme. Sec. 3.3 then introduces the overall loss functions.

3.2 Degradation-Driven Binning

As shown in Fig. 2, in addition to the LR depth $\mathbf{X}_1$, the context feature $\mathbf{Y}$ and the layer feature $\mathbf{G}_i$ extracted by the pretrained backbone $\mathcal{F}(\cdot)$ [46], we further estimate the degradation relationship [1] between the coarse depth $\mathbf{X}_i$ and the final prediction $\mathbf{X}_5$ using the pretrained degradation learning

[1] Based on the assumption that a degradation relationship exists between the low-quality depth $\mathbf{X}$ and target depth $\hat{\mathbf{X}}$, defined as $\mathbf{X} = \mathbf{K}\hat{\mathbf{X}} + \mathbf{n}$, where $\mathbf{K}$ is a degradation kernel derived from the degradation representation $\mathbf{D}$ and $\mathbf{n}$ denotes noise, DORNet models $\mathbf{D}$ in a self-supervised manner by minimizing the discrepancy between $\mathbf{X}$ and $\mathbf{K}\hat{\mathbf{X}}$.

block $\mathcal{D}(\cdot)$ from DORNet [39]. This process is formulated as:

$$\mathbf{Y}, \mathbf{G}_i = \mathcal{F}(\mathbf{I}, \mathbf{X}_1), \tag{1a}$$

$$\mathbf{D}_i = \mathcal{D}(\mathbf{X}_i, \mathbf{X}_5), \quad \text{for} \quad i = 1, 2, 3, 4. \tag{1b}$$

Given these three inputs, the subsequent description of DDB is organized into three aspects: **(1)** decomposing depth into bin representations, **(2)** adaptively adjusting bin ranges and optimizing the probability distribution under degradation guidance, and **(3)** leveraging the former depth estimation to progressively update the subsequent prediction.

(1) In the i-th stage, DDB first discretizes the continuous coarse depth $\mathbf{X}_i$ into multiple intervals [1, 2, 26]. Let $v_{min}(p)$ and $v_{max}(p)$ denote the minimum and maximum of $\mathbf{X}_i$ at pixel p, respectively. Then, the pixel-wise depth range $[v_{min}(p), v_{max}(p)]$ is uniformly divided into N bins, yielding the width $W(p)$ of each bin and the upper bound $u_n(p)$ of the n-th bin:

$$W(p) = \frac{v_{max}(p) - v_{min}(p)}{N}, \tag{2a}$$

$$u_n(p) = v_{min}(p) + nW(p), \quad \text{for} \quad n = 0, 1, \cdots, N; \tag{2b}$$

where the minimum and maximum values adaptively vary with pixels of the coarse depth, and N is set to 32 unless otherwise specified. Since discrete bins cannot be directly used for depth prediction, their center values are adopted as the corresponding depth candidates:

$$C_n^{'}(p) = \frac{u_n(p) + u_{n+1}(p)}{2}, \quad \text{for} \quad n = 0, 1, \cdots, N-1. \tag{3}$$

Based on these bin centers, weighted by their learned probability distribution $\mathbf{P}_n^{'}$ (updated by Eq. (9)), the depth at the pixel p in $\mathbf{X}_i^{'}$ is computed as:

$$\mathbf{X}_i^{'}(p) = \sum_{i=0}^{N-1} C_n^{'}(p) \cdot \mathbf{P}_n^{'}(p), \tag{4}$$

where $\mathbf{P}_n^{'}$ denotes the n-th depth probability.

(2) Bin Range Adjustment. The target bin is defined as the interval whose boundaries enclose the predicted depth value, *i.e.*, the bin for which $\mathbf{X}_i^{'}(p)$ falls within its edges:

$$u_n(p) \leq \mathbf{X}_i^{'}(p) \leq u_{n+1}(p) \tag{5}$$

which is denoted as $\mathbf{X}_i^{'}(p) \in [u_n(p), u_{n+1}(p)]$. As illustrated in the yellow region of Fig. 2, we leverage the degradation representation $\mathbf{D}_i$, which characterizes the likelihood of depth inaccuracy, to adaptively adjust this interval. Motivated by [11, 42, 26], we calculate the variance of the degradation:

$$V(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} \left(\mathbf{D}(q) - \mu(p)\right)^2, \tag{6}$$

where $\mathcal{N}(p)$ denotes a $k \times k$ local neighborhood (with $k = 3$ by default) centered at pixel p, and $\mu(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} \mathbf{D}(q)$ denotes the corresponding local mean. The standard deviation is then computed as $\sigma(p) = \sqrt{V(p)}$. Given the target bin range $[u_n(p), u_{n+1}(p)]$ and the standard deviation $\sigma(p)$, we can acquire the new range:

$$[u_n(p) - \gamma\sigma(p), u_{n+1}(p) + \gamma\sigma(p)], \tag{7}$$

where γ is a coefficient controlling the error tolerance and is set to 0.2. Thus we can renew $v_{min}(p)$ and $v_{max}(p)$ as $u_n(p) - 0.2\sigma(p)$ and $u_{n+1}(p) + 0.2\sigma(p)$, yielding updated depth candidate C_n and prediction via Eq. (4). For pixels with large discrepancies from the ground-truth (GT) depth, the standard deviation increases accordingly, yielding a more tolerant target bin with higher immunity that helps mitigate errors.

Probability Optimization. Unlike prior methods [1, 2, 18] that rely on heavy transformer modules, we efficiently estimate the probability distribution using a gated recurrent unit [32, 26] (GRU), which propagates contextual information from previous states. Specifically, we first project the depth candidates C into a feature space using four 3×3 convolutions, each followed by a ReLU activation.

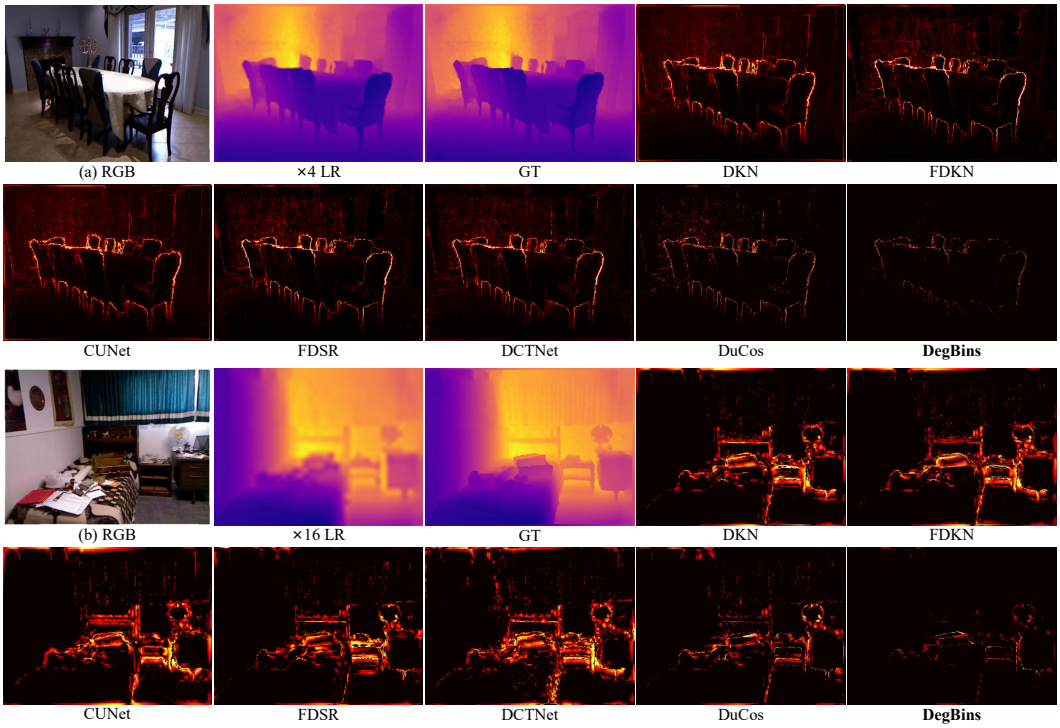


Figure 3: Error map comparisons on the synthetic NYU v2 with $\times 4$ and $\times 16$ upscaling factors.

The resulting feature is then concatenated with the context feature $\mathbf{Y}$ to form one of the GRU input $\mathbf{Q}$. Another key input, namely the hidden state $\mathbf{H}^{'t}$, is generated through a 3×3 convolution followed by a $\tanh$ activation. We further incorporate the degradation priority into this hidden representation using a deformable convolutional network [63] (DCN) $\mathcal{J}(\cdot)$, where both the offsets and modulation scalars are derived from the degradation $\mathbf{D}_i$, yielding a new hidden state:

$$\mathbf{H}^{t} = \mathbf{H}^{'t} + \mathcal{J}(\mathbf{H}^{'t}, \mathbf{D}_i). \tag{8}$$

Given the input $\mathbf{Q}$ and the updated hidden state $\mathbf{H}^{t}$, the hidden state at the next time step is computed as $\mathbf{H}^{t+1} = \mathcal{G}(\mathbf{Q}, \mathbf{H}^{t})$. Then we obtain the probability via:

$$\mathbf{P} = \mathcal{F}_r(\mathcal{F}_t(\mathbf{H}^{t+1})), \tag{9}$$

where $\mathcal{F}_r(\cdot)$ and $\mathcal{F}_t(\cdot)$ denote the ReLU and softmax activation functions, respectively. Consequently, by incorporating Eq. (7) and Eq. (9), Eq. (4) can be reformulated as:

$$\mathbf{X}_i^{'}(p) = \sum_{i=0}^{N-1} C_n(p) \cdot \mathbf{P}_n(p). \tag{10}$$

(3) We summarize all the preceding modeling in this section as a unified DDB process, where the inputs are the coarse depth $\mathbf{X}_i$, context feature $\mathbf{Y}$, layer feature $\mathbf{G}_i$, and degradation $\mathbf{D}_i$, and the output is $\mathbf{X}_{i+1}$. Denoting the DDB process function as $\mathcal{P}(\cdot)$, the formulation for the progressive scheme is given as follows:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathcal{P}(\mathbf{X}_i, \mathbf{G}_i, \mathbf{D}_i, \mathbf{Y}) + \mathbb{I}_{\{i=4\}}\, \mathbf{X}_1'. \tag{11}$$

The depth is gradually refined based on its previous estimation. In short, we aim to mitigate the limitations of residual-based DSR models in capturing the complex relationship between HR and LR by leveraging a DDB strategy built upon high-dimensional degradation representations.

3.3 Loss Function

Given the set of bin centers as $\mathcal{C}$, HR depth $\mathbf{X}$, and GT depth $\mathbf{Z}$ with m valid pixels, the reconstruction loss function is written as $\mathcal{L}_{rec} = \frac{1}{m}\|\mathbf{X} - \mathbf{Z}\|_1$. Following [1, 2], we use the bi-directional

Table 1: Results on synthetic DSR benchmarks. The best and second-best metrics are highlighted.

Method	Scale	Middlebury			Lu			NYU v2			RGB-D-D			TOFDSR		
		RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1
DJF	×2	1.14	0.63	98.56	1.08	0.44	99.00	2.32	0.83	99.49	1.18	0.36	99.64	2.99	0.66	98.74
DJFR		1.22	0.58	98.42	1.39	0.45	98.94	1.87	0.51	99.68	0.98	0.27	99.74	1.58	0.28	99.57
CUNet		1.01	0.58	98.78	1.06	0.50	99.10	1.50	0.46	99.81	0.84	0.25	99.81	1.69	0.33	99.49
FDKN		1.35	0.62	98.39	1.59	0.49	98.91	2.05	0.50	99.63	1.04	0.26	99.70	1.85	0.36	99.32
DKN		1.32	0.62	98.48	1.48	0.48	99.00	1.95	0.49	99.66	0.98	0.24	99.74	1.87	0.36	99.33
FDSR		0.94	0.51	98.89	0.94	0.39	99.14	1.74	0.57	99.70	0.91	0.26	99.76	1.55	0.40	99.24
DCTNet		1.01	0.53	98.90	1.03	0.59	99.36	1.59	0.59	99.76	0.89	0.28	99.79	0.80	0.23	99.87
DADA		1.28	0.60	98.47	1.45	0.48	99.07	2.00	0.54	99.65	1.04	0.27	99.71	1.73	0.37	99.26
DuCos		0.81	0.46	99.28	0.65	0.26	99.68	1.20	0.37	99.88	0.74	0.22	99.85	0.52	0.13	99.94
DegBins		0.76	0.43	99.51	0.57	0.22	99.74	1.12	0.30	99.94	0.68	0.19	99.90	0.46	0.09	99.98
DJF	×4	1.93	1.11	97.11	1.93	1.11	98.20	3.60	1.67	99.05	1.63	0.74	99.39	3.75	1.14	98.18
DJFR		1.83	1.04	96.91	1.85	1.02	98.36	3.27	1.14	99.04	1.54	0.48	99.35	2.97	0.58	99.00
CUNet		1.61	0.98	97.79	1.73	0.97	98.48	3.22	1.44	99.21	1.52	0.65	99.43	3.57	1.13	98.19
FDKN		1.80	0.80	97.69	2.11	0.58	98.53	3.28	0.93	99.28	1.56	0.42	99.42	2.80	0.58	98.96
DKN		1.77	0.78	97.77	2.05	0.56	98.58	3.15	0.90	99.34	1.50	0.41	99.49	2.73	0.56	99.04
FDSR		1.72	0.84	97.69	1.95	0.56	98.52	2.94	0.85	99.39	1.41	0.40	99.51	2.41	0.51	99.07
DCTNet		1.66	0.77	97.86	1.85	0.57	98.56	2.90	0.99	99.37	1.49	0.44	99.45	2.86	0.59	98.98
DADA		1.82	0.83	97.45	2.12	0.66	98.35	3.08	0.96	99.33	1.63	0.46	99.39	2.85	0.68	98.57
DORNet		1.69	0.80	97.76	1.78	0.56	98.70	2.79	0.85	99.44	1.45	0.41	99.50	2.28	0.36	99.49
DuCos		1.45	0.68	98.30	1.38	0.41	99.08	2.60	0.81	99.49	1.27	0.36	99.60	2.09	0.33	99.58
DegBins		1.36	0.64	98.42	1.33	0.39	99.14	2.57	0.78	99.51	1.25	0.32	99.65	1.97	0.30	99.64
DJF	×8	3.09	1.46	94.10	3.58	1.22	95.60	5.56	2.30	97.70	2.58	0.96	98.29	5.59	1.71	96.29
DJFR		2.82	1.25	95.31	3.24	1.07	95.86	5.20	1.94	98.21	2.61	0.93	98.25	5.11	1.35	97.16
CUNet		2.86	1.46	94.44	2.85	1.25	96.32	5.50	2.23	97.78	2.35	0.92	98.60	5.14	1.64	96.67
FDKN		2.51	1.13	96.26	2.67	0.85	97.39	4.93	1.67	98.65	2.25	0.70	98.86	4.40	0.96	98.15
DKN		2.43	1.12	96.26	2.88	0.88	97.13	4.88	1.71	98.62	2.33	0.74	98.76	4.54	1.06	97.87
FDSR		2.41	1.13	96.24	2.69	0.86	97.35	4.82	1.69	98.57	2.25	0.73	98.72	4.28	0.95	98.06
DCTNet		2.75	1.31	95.25	3.07	1.18	96.38	4.90	2.12	98.28	2.47	0.92	98.60	5.38	1.57	97.01
DADA		2.77	1.27	95.01	3.76	1.17	96.01	4.83	1.86	98.34	2.81	0.93	98.29	5.86	1.64	96.15
DORNet		2.34	1.03	96.85	2.81	0.82	97.50	4.76	1.58	98.81	2.33	0.69	98.80	4.81	0.97	98.14
DuCos		2.23	0.97	97.07	2.67	0.72	97.86	4.61	1.52	98.86	2.23	0.66	98.89	4.60	0.92	98.30
DegBins		2.20	0.96	97.12	2.63	0.68	97.99	4.57	1.45	98.87	2.14	0.61	98.94	4.42	0.86	98.40
DJF	×16	5.50	2.92	84.66	6.53	2.69	88.91	9.82	4.73	93.05	4.46	2.04	94.99	8.19	3.49	90.79
DJFR		5.16	2.61	86.39	6.46	2.38	89.66	9.50	4.28	93.63	4.36	1.84	95.48	8.06	3.06	92.20
CUNet		4.72	2.40	88.45	5.63	2.17	91.33	8.63	3.88	94.55	3.81	1.58	96.36	7.36	2.73	93.40
FDKN		4.42	2.10	90.80	5.48	1.91	92.24	7.97	3.43	95.76	3.71	1.45	96.81	7.16	2.29	94.78
DKN		4.17	1.97	91.67	5.44	1.90	91.94	7.70	3.28	96.06	3.70	1.39	97.06	7.24	2.20	95.26
FDSR		3.97	1.81	92.52	5.23	1.67	93.44	7.29	2.91	96.86	3.44	1.24	97.37	6.85	1.91	95.96
DCTNet		5.07	2.59	86.91	5.83	2.25	90.19	9.10	4.24	94.04	4.13	1.66	96.30	8.04	2.77	93.45
DADA		4.11	2.06	90.09	6.19	2.22	90.95	7.99	3.54	95.64	4.01	1.59	96.71	7.79	2.70	93.25
DORNet		4.32	1.98	91.78	5.27	1.75	93.21	7.60	3.00	96.81	3.88	1.34	97.23	7.94	2.19	95.46
DuCos		3.96	1.69	93.52	5.18	1.59	93.89	7.37	2.82	97.12	3.44	1.16	97.61	6.90	1.77	96.51
DegBins		3.74	1.60	94.02	4.93	1.48	94.27	7.11	2.65	97.36	3.29	1.10	97.82	6.73	1.68	96.97
DJF	×32	9.14	5.33	70.35	11.03	5.16	77.59	16.14	8.29	84.68	7.29	3.85	87.33	12.29	6.56	82.85
DJFR		8.26	4.79	72.45	9.98	4.62	78.98	15.22	7.76	85.77	6.60	3.55	89.20	11.48	5.76	85.03
CUNet		8.48	4.87	72.75	10.56	4.90	78.61	15.42	7.84	85.64	6.90	3.56	89.13	11.81	6.01	84.69
FDKN		7.73	4.40	75.11	9.24	4.37	79.71	14.06	7.20	87.17	6.30	3.24	90.76	10.74	5.13	86.99
DKN		7.35	4.14	76.97	9.50	4.59	78.94	13.96	7.10	87.47	6.12	3.05	91.29	10.46	4.80	87.47
FDSR		7.79	4.31	76.11	9.15	4.21	80.35	13.55	6.96	87.71	6.05	3.00	91.32	9.64	4.38	88.40
DCTNet		9.02	5.22	70.59	10.45	5.18	76.27	15.99	8.51	83.93	6.45	3.42	89.72	10.87	5.39	85.81
DADA		6.84	3.75	78.99	9.27	4.44	79.43	12.94	6.74	88.40	5.52	2.84	92.37	9.60	4.49	88.22
DORNet		7.56	4.25	76.98	9.15	4.42	79.65	13.81	6.92	88.25	5.77	2.72	92.50	10.14	4.14	89.62
DuCos		7.69	4.14	78.63	10.40	4.68	78.56	12.27	5.60	92.00	4.90	2.03	95.40	9.22	3.03	93.30
DegBins		6.87	3.70	81.15	9.12	3.93	81.61	12.06	5.47	92.34	4.84	1.99	95.50	9.16	2.98	93.37

Chamfer loss [6] as a binning regularizer: $\mathcal{L}_{bin} = \sum_{z \in \mathbf{Z}} \min_{c \in \mathcal{C}} \|c - z\|^2 + \sum_{c \in \mathcal{C}} \min_{z \in \mathbf{Z}} \|c - z\|^2$. Consequently, the total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha \mathcal{L}_{bin}, \tag{12}$$

where α denotes a balance coefficient that is empirically set to 0.1.

4 Experiment

Dataset. Following DuCos [46], all methods are trained on the fully simulated Hypersim [23] dataset and evaluated on test splits of multiple unseen DSR benchmarks, including TOFDSR [43, 45], RGB-D-D [9], NYU v2 [28], Middlebury [10, 24], and Lu [21]. Consistent with prior works [9, 56, 46], we construct LR depth inputs via bicubic downsampling of GT depth annotations across these datasets. In addition, TOFDSR and RGB-D-D further provide real-world LR depth measurements, allowing for a more comprehensive evaluation of model generalization under real sensing conditions. Refer to our Appendix A for metrics and implementation details.

4.1 Comparison with State-of-the-Arts

In this section, we compare DegBins with various methods: DJF [16], DJFR [17], CUNet [4], DKN [13], FDKN [13], FDSR [9], DCTNet [55], SUFT [27], SFG [50], DADA [22], DORNet [39],

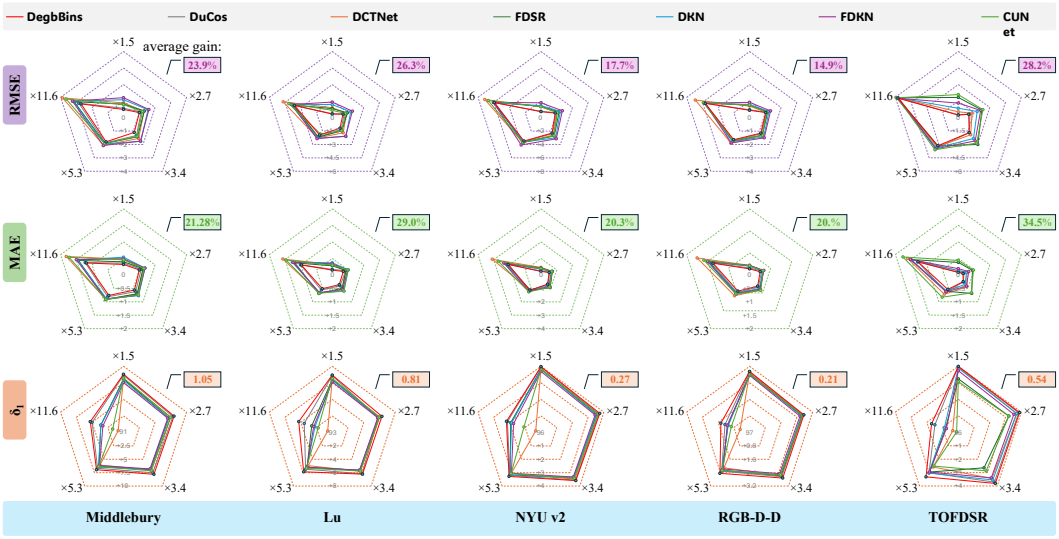


Figure 4: Quantitative comparisons of arbitrary scaling factors on five synthetic DSR datasets.

and DuCos [46]. We further evaluate their generalization performance across five settings: synthetic DSR, arbitrary-scale DSR, cross-scale DSR, real-world DSR, and compressed DSR.

Synthetic DSR. Tab. 1 reports quantitative comparisons across five scaling factors on multiple public synthetic datasets. It is evident that our DegBins consistently achieves superior effectiveness and robustness on all evaluation metrics. For small scaling factors ($\times 2$ and $\times 4$), DegBins obtains the lowest RMSE and MAE while maintaining δ_1 accuracy near or above 99%, highlighting its advantage in both detail restoration and pixel-wise precision. For example, compared to the suboptimal method DuCos, our DegBins reduces RMSE by $0.08cm$ and MAE by $0.07cm$ on the $\times 2$ NYU v2 dataset. As the scaling factor increases, reconstructing high-quality HR depth becomes more challenging. Nevertheless, our method delivers even larger performance gains, surpassing the second-best approach by $0.18cm$ and $0.21cm$ in RMSE on $\times 16$ and $\times 32$ NYU v2 dataset, respectively. Furthermore, Fig. 3 presents visual comparisons across different scaling factors, confirming that DegBins successfully recovers accurate depth and with fewer prediction errors.

Arbitrary-scale DSR. In real-world applications, scaling factors are often non-fixed, including both integer and non-integer values. To evaluate the generalization ability of our method under arbitrary scales, we further conduct experiments with multiple non-integer factors. As shown in Fig. 4, DegBins consistently delivers highly competitive performance across all datasets, metrics, and scales. For instance, compared with all competing methods, DegBins achieves an average improvement of 23.9% in RMSE, 21.28% in MAE, and 1.05 percent points in δ_1 on the Middlebury dataset.

Cross-scale DSR. To validate the generalization of our method under large scale factor discrepancies, Tab. 2 further presents cross-scale quantitative comparisons, where the model weights trained on the $\times 4$ Hypersim dataset are directly applied to evaluate $\times 8$ and $\times 16$ DSR tasks without any fine-tuning. Compared with existing methods, our approach exhibits satisfactory cross-scale robustness. For example, DegBins achieves an average RMSE reduction of $0.17cm$ over DuCos, DORNet, SUFT, and DCTNet on the synthetic RGB-D-D dataset ($\times 4 \rightarrow \times 8$). These results strongly demonstrate the effectiveness and superiority of our DegBins in handling scenes with large scale variations.

Table 2: Cross-scale DSR on synthetic RGB-D-D and Lu.

Method	scale	RGB-D-D			Lu		
		RMSE	MAE	$\delta_{1.05}$	RMSE	MAE	$\delta_{1.05}$
CUNet	$\times 4 \rightarrow \times 8$	3.30	1.66	97.07	4.90	2.59	93.74
FDKN		3.39	1.22	96.97	4.94	1.56	94.03
DKN		3.31	1.22	96.97	4.86	1.52	94.21
FDSR		3.29	1.19	97.01	4.81	1.54	94.18
DCTNet		3.33	1.20	97.03	4.92	1.55	94.14
SUFT		3.24	1.20	97.04	4.79	1.64	94.08
DORNet		3.22	1.18	97.13	4.71	1.52	94.23
DuCos		3.19	1.17	97.11	4.64	1.46	94.28
DegBins		3.15	1.14	97.18	4.60	1.42	94.33
CUNet	$\times 4 \rightarrow \times 16$	5.25	2.69	92.89	7.79	3.85	87.19
FDKN		5.16	2.34	92.97	7.56	2.93	87.91
DKN		5.15	2.32	93.00	7.54	2.92	87.93
FDSR		5.16	2.33	93.01	7.55	2.92	87.93
DCTNet		5.15	2.32	92.99	7.55	2.91	88.02
SUFT		5.14	2.32	93.01	7.55	2.98	87.96
DORNet		5.14	2.32	93.02	7.53	2.91	87.99
DuCos		5.13	2.31	93.03	7.51	2.87	88.08
DegBins		5.06	2.28	93.12	7.47	2.85	88.14

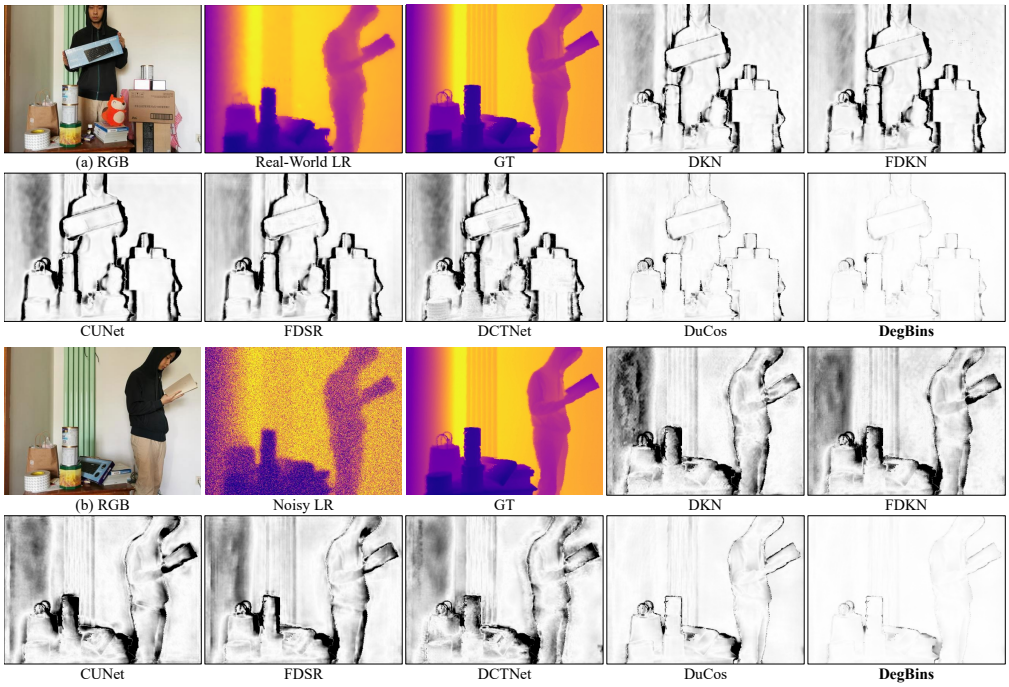


Figure 5: Error map comparisons on the real-world RGB-D-D dataset.

Table 3: Results on real-world DSR datasets: RGB-D-D, TOFDSR, and their noisy patterns.

Method	RGB-D-D			TOFDSR			RGB-D-D w/ noise			TOFDSR w/ noise		
	RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1
DJF	5.54	3.43	93.81	5.84	2.13	96.79	7.94	5.16	84.82	11.45	7.87	67.95
DJFR	5.52	3.51	93.58	5.72	2.10	97.03	7.50	4.83	86.25	10.92	7.39	70.46
CUNet	5.84	3.06	94.75	6.04	2.21	96.46	6.69	4.14	89.36	9.76	5.86	80.43
FDKN	5.37	2.70	96.05	5.77	2.19	97.33	6.66	4.26	90.09	8.13	4.66	86.24
DKN	5.08	2.58	96.28	5.50	2.07	97.54	6.50	4.16	90.04	7.42	4.29	88.20
FDSR	5.49	3.10	94.77	5.03	1.67	97.61	6.39	4.07	90.69	6.31	3.17	92.74
DCTNet	5.43	3.29	93.15	5.16	2.10	96.37	6.04	3.79	90.90	7.52	4.50	86.04
SFG	3.88	1.96	97.09	4.52	1.72	97.45	5.87	3.79	89.84	5.46	2.89	93.02
DuCos	3.68	1.54	97.87	4.29	1.15	98.67	4.14	2.00	96.76	5.20	2.20	96.44
DegBins	3.52	1.45	98.01	4.06	1.03	98.87	3.90	1.91	96.94	5.04	2.11	96.59

Real-world DSR. Unlike synthetic data, real-world LR depth often suffers from unknown and complex degradations, such as noise, holes, and structural distortions, posing significant challenges for real-world DSR applications. The first two columns of Tab. 3 report the results on the RGB-D-D and TOFDSR datasets, where our method consistently outperforms all competing approaches. On TOFDSR dataset, our DegBins reduces RMSE by $0.23cm$ and MAE by $0.12cm$ compared to the sub-optimal DuCos. Furthermore, the last two columns of Tab. 3 present the results of joint denoising and DSR, where Gaussian blur (standard deviation 3.6) and Gaussian noise (mean 0 and standard deviation 0.07) are added to the LR depth as new LR inputs. These experimental results highlight the strong noise robustness of our DegBins. For instance, compared to the second-best method, our DegBins reduces RMSE by $0.24cm$ on the RGB-D-D dataset and $0.16cm$ on the TOFDSR dataset, respectively. Fig. 5 visualizes the prediction results, further confirming that DegBins can recover sharp and accurate depth under real-world degradations.

Table 4: Results of compressed DSR on NYU v2, RGB-D-D, and TOFDSR.

Method	NYU v2			RGB-D-D			TOFDSR		
	RMSE	MAE	δ_1	RMSE	MAE	δ_1	RMSE	MAE	δ_1
DJF	119.41	101.14	8.53	34.48	29.93	16.26	47.98	44.30	10.94
DJFR	147.43	121.73	7.85	40.46	34.49	14.03	50.14	45.81	8.59
CUNet	133.90	108.96	7.14	35.03	30.67	15.08	50.25	46.53	8.96
FDKN	187.53	154.15	4.26	42.37	36.91	14.04	49.34	44.88	14.28
DKN	186.10	152.74	4.33	40.38	35.15	14.40	47.27	43.34	9.58
FDSR	118.76	101.28	8.39	35.88	30.15	20.64	45.96	40.92	20.26
DCTNet	118.45	101.30	8.20	37.85	33.25	12.82	51.58	48.41	6.88
DORNet	128.25	106.11	8.58	41.13	35.07	15.28	53.77	47.22	12.92
DuCos	58.74	46.00	26.19	24.64	21.13	26.45	30.40	25.63	22.37
DegBins	51.26	43.48	28.81	19.74	18.41	30.63	27.35	24.52	22.80

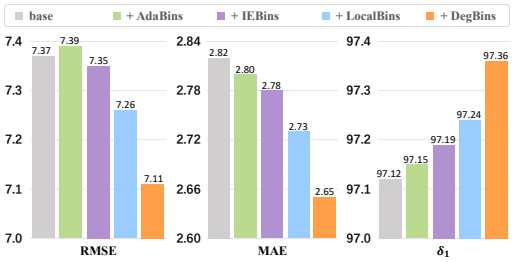


Figure 6: Ablation of different bin strategies on the synthetic NYU v2 ($\times$16) dataset.

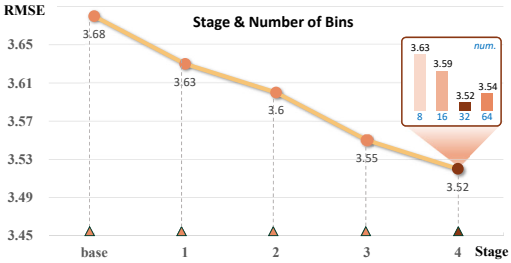


Figure 7: Ablation of the stage and number of bins on the real-world RGB-D-D dataset.

Table 5: Ablation of the impact of degradation on synthetic NYU v2 and real-world RGB-D-D.

Method	Deg. in Bins	Deg. in Prob. Dist.	NYU v2			RGB-D-D		
			RMSE	MAE	δ_1	RMSE	MAE	δ_1
i			7.37	2.82	97.12	3.68	1.54	97.87
ii	✓		7.28	2.76	97.19	3.63	1.50	97.94
iii		✓	7.20	2.74	97.25	3.59	1.48	97.96
iv	✓	✓	7.11	2.65	97.36	3.52	1.45	98.01

Compressed DSR. Compression is often necessary for efficient depth transmission, particularly in real-time downstream applications. However, the compression inevitably introduces artifacts and degrades depth quality, making accurate HR depth reconstruction from compressed LR depth crucial for real-world deployment. Following [57, 46], we evaluate DegBins on the compressed DSR task. Tab. 4 shows that DegBins consistently achieves competitive performance on all datasets, surpassing the suboptimal method by $7.48cm$, $4.90cm$, and $3.05cm$ in RMSE on NYU v2, RGB-D-D, and TOFDSR, respectively. These results validate the superiority of DegBins in compressed scenarios.

4.2 Ablation Study

The baseline removes all bin-related modules from DegBins, retaining only the gray branch in Fig. 2.

Bin Types. Fig. 6 compares our DegBins with previous bin-based methods, including AdaBins [1], IEBins [26], and LocalBins [2]. To ensure a fair comparison, we directly replace our DegBins strategy with each of these methods while keeping the rest of the network architecture unchanged. These experimental results demonstrate that our DegBins achieves the best performance across all evaluation metrics. On average, our method outperforms the competing approaches by $0.23cm$ in RMSE, $0.12cm$ in MAE, and 0.17 percentage points in δ_1. These results confirm that our approach effectively reduces DSR error and improves reconstruction quality.

DDB Details. As illustrated in the line chart of Fig. 7, we conduct an ablation analysis with different numbers of stages. Here, '2-Stage' indicates that our method only employs the 'Layers' from the first two stages along with their corresponding DDBs (see Fig. 2). It can be clearly observed that as the number of stages increases, the RMSE consistently decreases. When all four stages are introduced, our DegBins achieves the best performance, surpassing the baseline by $0.16cm$ RMSE on the real-world RGB-D-D dataset. Therefore, we adopt four stages as the default setting. In addition, the bar chart in Fig. 7 presents a quantitative comparison for different numbers of bins. These results show that DSR performance gradually improves as the number of bins increases, while it begins to decline once the number exceeds 32. As a result, we adopt 32 bins for DegBins.

Furthermore, Tab. 5 reports the impact of degradation representations on bin construction. Compared with Baseline (i), the results show that introducing degradation representations enables adaptive modulation of both bin ranges (ii) and probability distributions (iii), significantly improving depth reconstruction accuracy. When modulating both components simultaneously, our method (iv) achieves optimal performance, outperforming the baseline by $0.26cm$ in RMSE on the $\times$16 NYU v2 dataset and by $0.16cm$ in RMSE on the real-world RGB-D-D dataset.

5 Conclusion

In this paper, we present DegBins, a novel and generalizable DSR framework based on residual modeling. By reformulating DSR as a hybrid regression-classification task, DegBins facilitates more flexible and expressive residual representations. Furthermore, we introduce a multi-stage degradation-driven binning strategy that leverages high-dimensional local degradation cues to adaptively adjust bin ranges and optimize their probability distributions. Extensive experiments validate the superiority of DegBins in terms of accuracy, robustness, and generalization ability.

Limitation. Although DegBins achieves strong performance, it incurs relatively high computational complexity (see Appendix A.3) due to the heavy feature extraction backbone. In future work, we plan to explore lightweight feature extraction architectures to enable real-time performance.

Broader Impact. This work improves the accuracy, robustness, and generalization of depth super-resolution, benefiting many downstream applications such as robotics, augmented reality, and 3D scene understanding, where precise geometric perception is essential.

Acknowledgments

This research / project is supported by the National Research Foundation (NRF) Singapore, under its NRF-Investigatorship Programme (Award ID. NRF-NRFI09-0008), and the Tier 2 grant MOET2EP20124-0015 from the Singapore Ministry of Education.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, pages 4009–4018, 2021.

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *ECCV*, pages 480–496. Springer, 2022.

[3] Riccardo De Lutio, Alexander Becker, Stefano D'Aronco, Stefania Russo, Jan D Wegner, and Konrad Schindler. Learning graph regularisation for guided super-resolution. In *CVPR*, pages 1979–1988, 2022.

[4] Xin Deng and Pier Luigi Dragotti. Deep convolutional neural network for multi-modal image restoration and fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (10):3333–3348, 2020.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 27, 2014.

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017.

[7] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *ICCV*, pages 993–1000, 2013.

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[9] Lingzhi He, Hongguang Zhu, Feng Li, Huihui Bai, Runmin Cong, Chunjie Zhang, Chunyu Lin, Meiqin Liu, and Yao Zhao. Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline. In *CVPR*, pages 9229–9238, 2021.

[10] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, pages 1–8, 2007.

[11] Dongting Hu, Liuhua Peng, Tingjin Chu, Xiaoxing Zhang, Yinian Mao, Howard Bondell, and Mingming Gong. Uncertainty quantification in depth estimation via constrained ordinal regression. In *ECCV*, pages 237–256. Springer, 2022.

[12] Jiahui Kang, Qing Cai, Runqing Tan, Yimei Liu, and Zhi Liu. C2pd: Continuity-constrained pixelwise deformation for guided depth super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4212–4220, 2025.

[13] Beomjun Kim, Jean Ponce, and Bumsub Ham. Deformable kernel networks for joint image filtering. *International Journal of Computer Vision*, 129(2):579–600, 2021.

[14] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *CVPR*, pages 17452–17462, 2022.

[16] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *ECCV*, pages 154–169, 2016.

[17] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1909–1923, 2019.

[18] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *IEEE Transactions on Image Processing*, 33:3964–3976, 2024.

[19] Jie Liang, Hui Zeng, and Lei Zhang. Efficient and degradation-adaptive network for real-world image super-resolution. In *ECCV*, pages 574–591, 2022.

[20] Kai-Han Lo, Yu-Chiang Frank Wang, and Kai-Lung Hua. Edge-preserving depth map upsampling by joint trilateral filter. *IEEE transactions on cybernetics*, 48(1):371–384, 2017.

[21] Si Lu, Xiaofeng Ren, and Feng Liu. Depth enhancement via low-rank matrix completion. In *CVPR*, pages 3390–3397, 2014.

[22] Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Guided depth super-resolution by deep anisotropic diffusion. In *CVPR*, pages 18237–18246, 2023.

[23] 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 *ICCV*, pages 10912–10922, 2021.

[24] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *CVPR*, pages 1–8, 2007.

[25] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *ICCV*, pages 7931–7940, 2023.

[26] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. Iebins: Iterative elastic bins for monocular depth estimation. *NeurIPS*, 36:53025–53037, 2023.

[27] Wuxuan Shi, Mang Ye, and Bo Du. Symmetric uncertainty-aware feature transmission for depth super-resolution. In *ACM MM*, pages 3867–3876, 2022.

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[29] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, pages 11166–11175, 2019.

[30] Baoli Sun, Xinchen Ye, Baopu Li, Haojie Li, Zhihui Wang, and Rui Xu. Learning scene structure guidance via cross-task knowledge transfer for single depth super-resolution. In *CVPR*, pages 7792–7801, 2021.

[31] Qi Tang, Runmin Cong, Ronghui Sheng, Lingzhi He, Dan Zhang, Yao Zhao, and Sam Kwong. Bridgenet: A joint learning network of depth map super-resolution and monocular depth estimation. In *ACM MM*, pages 2148–2157, 2021.

[32] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020.

[33] Haotian Wang, Meng Yang, and Nanning Zheng. G2-monodepth: A general framework of generalized depth inference from monocular rgb+ x data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3753–3771, 2023.

[34] Kun Wang, Zhiqiang Yan, Junkai Fan, Wanlu Zhu, Xiang Li, Jun Li, and Jian Yang. Dcdepth: Progressive monocular depth estimation in discrete cosine domain. *NeurIPS*, 37:64629–64648, 2024.

[35] Kun Wang, Yun Zhu, Pan Zhou, and Na Zhao. Robust depth super-resolution via adaptive diffusion sampling. *arXiv preprint arXiv:2602.09510*, 2026.

[36] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *CVPR*, pages 10581–10590, 2021.

[37] Zhengxue Wang, Zhiqiang Yan, and Jian Yang. Sgnet: Structure guided network via gradient-frequency awareness for depth map super-resolution. In *AAAI*, pages 5823–5831, 2024.

[38] Zhengxue Wang, Zhiqiang Yan, Ming-Hsuan Yang, Jinshan Pan, Jian Yang, Ying Tai, and Guangwei Gao. Scene prior filtering for depth map super-resolution. *arXiv preprint arXiv:2402.13876*, 2024.

[39] Zhengxue Wang, Zhiqiang Yan, Jinshan Pan, Guangwei Gao, Kai Zhang, and Jian Yang. Dornet: A degradation oriented and regularized network for blind depth super-resolution. In *CVPR*, pages 15813–15822, 2025.

[40] Zhengxue Wang, Yuan Wu, Xiang Li, Zhiqiang Yan, and Jian Yang. Spatiotemporal difference network for video depth super-resolution. In *AAAI*, pages 10403–10411, 2026.

[41] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Guangyu Li, Jun Li, and Jian Yang. Learning complementary correlations for depth super-resolution with incomplete data in real world. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5616–5626, 2022.

[42] Zhiqiang Yan, Xiang Li, Kun Wang, Shuo Chen, Jun Li, and Jian Yang. Distortion and uncertainty aware loss for panoramic depth completion. In *ICML*, pages 39099–39109, 2023.

[43] Zhiqiang Yan, Yuankai Lin, Kun Wang, Yupeng Zheng, Yufei Wang, Zhenyu Zhang, Jun Li, and Jian Yang. Tri-perspective view decomposition for geometry-aware depth completion. In *CVPR*, pages 4874–4884, 2024.

[44] Zhiqiang Yan, Zhengxue Wang, Kun Wang, Jun Li, and Jian Yang. Completion as enhancement: A degradation-aware selective image guided network for depth completion. *arXiv preprint arXiv:2412.19225*, 2024.

[45] Zhiqiang Yan, Kun Wang, Xiang Li, Guangwei Gao, Jun Li, and Jian Yang. Tri-perspective view decomposition for geometry aware depth completion and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[46] Zhiqiang Yan, Zhengxue Wang, Haoye Dong, Jun Li, Jian Yang, and Gim Hee Lee. Ducos: Duality constrained depth super-resolution via foundation model. In *ICCV*, pages 8361–8371, 2025.

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

[48] Mingde Yao, Ruikang Xu, Yuanshen Guan, Jie Huang, and Zhiwei Xiong. Neural degradation representation learning for all-in-one image restoration. *IEEE Transactions on Image Processing*, 2024.

[49] Guanghao Yin, Wei Wang, Zehuan Yuan, Wei Ji, Dongdong Yu, Shouqian Sun, Tat-Seng Chua, and Changhu Wang. Conditional hyper-network for blind super-resolution with multiple degradations. *IEEE Transactions on Image Processing*, 31:3949–3960, 2022.

[50] Jiayi Yuan, Haobo Jiang, Xiang Li, Jianjun Qian, Jun Li, and Jian Yang. Structure flow-guided network for real depth super-resolution. In *AAAI*, pages 3340–3348, 2023.

[51] Cheng Zhang, Yu Zhu, Qingsen Yan, Jinqiu Sun, and Yanning Zhang. All-in-one multi-degradation image restoration network via hierarchical degradation representation. In *ACM MM*, pages 2285–2293, 2023.

[52] Jinghao Zhang, Jie Huang, Mingde Yao, Zizheng Yang, Hu Yu, Man Zhou, and Feng Zhao. Ingredient-oriented multi-degradation learning for image restoration. In *CVPR*, pages 5825–5835, 2023.

[53] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, pages 4791–4800, 2021.

[54] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018.

[55] Zixiang Zhao, Jiangshe Zhang, Shuang Xu, Zudi Lin, and Hanspeter Pfister. Discrete cosine transform network for guided depth map super-resolution. In *CVPR*, pages 5697–5707, 2022.

[56] Zixiang Zhao, Jiangshe Zhang, Xiang Gu, Chengli Tan, Shuang Xu, Yulun Zhang, Radu Timofte, and Luc Van Gool. Spherical space feature decomposition for guided depth map super-resolution. In *ICCV*, pages 12547–12558, 2023.

[57] Huan Zheng, Wencheng Han, and Jianbing Shen. Decoupling fine detail and global geometry for compressed depth map super-resolution. In *CVPR*, pages 951–960, 2025.

[58] Zhiwei Zhong, Xianming Liu, Junjun Jiang, Debin Zhao, Zhiwen Chen, and Xiangyang Ji. High-resolution depth maps imaging via attention-based hierarchical multi-modal fusion. *IEEE Transactions on Image Processing*, 31:648–663, 2021.

[59] Zhiwei Zhong, Xianming Liu, Junjun Jiang, Debin Zhao, and Xiangyang Ji. Deep attentional guided image filtering. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[60] Zhiwei Zhong, Xianming Liu, Junjun Jiang, Debin Zhao, and Xiangyang Ji. Guided depth map super-resolution: A survey. *ACM Computing Surveys*, 55(14s):1–36, 2023.

[61] Zhiwei Zhong, Xianming Liu, Junjun Jiang, Debin Zhao, and Shiqi Wang. Dual-level cross-modality neural architecture search for guided image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[62] Hongyang Zhou, Xiaobin Zhu, Jianqing Zhu, Zheng Han, Shi-Xue Zhang, Jingyan Qin, and Xu-Cheng Yin. Learning correction filter via degradation-adaptive regression for blind single image super-resolution. In *ICCV*, pages 12365–12375, 2023.

[63] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019.

A Technical Appendices for DegBins

A.1 Metrics

Given the predicted depth map $\mathbf{X}$ and the corresponding GT $\mathbf{Z}$ containing m valid pixels, we evaluate performance using RMSE (cm), MAE (cm), and the accuracy metric δ_1, defined as follows:

$$\begin{aligned} &\text{MAE}: \frac{1}{m}\sum|\mathbf{Z}-\mathbf{X}|, \\ &\text{RMSE}: \sqrt{\frac{1}{m}\sum(\mathbf{Z}-\mathbf{X})^2}, \\ &\delta_i: \frac{r}{m}\times 100\%,\ r: \max(\mathbf{Z}/\mathbf{X}, \mathbf{X}/\mathbf{Z}) < 1.05^i. \end{aligned} \tag{13}$$

Since the values of δ_2 and δ_3 are typically very close across different DSR methods, we primarily report δ_1 for comparison.

A.2 Implementation Details

DegBins is implemented in PyTorch and trained on four NVIDIA RTX A6000 GPUs. The model is optimized for 200 epochs using the Adam optimizer [14], with an initial learning rate of 5×10^{-5}, which is reduced by a factor of 2 at epochs 40, 80, and 120. For data augmentation, we adopt random horizontal flipping and random 90° rotations to improve robustness. Since Hypersim [23] provides 1024×768 high-resolution images, while the test sets of other benchmarks are typically of much lower resolution, we randomly crop 256×256 patches during training on Hypersim to ensure efficiency and improve generalization.

A.3 Complexity Analysis

Table 6: Complexity comparisons on real-world RGB-D-D and synthetic NYU v2 ($\times$16) datasets using a Single RTX 4090 GPU.

Method	Params. (M)	Time (ms)	Speed (FPS)	RMSE (cm)	
				RGB-D-D	NYU v2
DCTNet [55]	0.48	9.15	109.29	5.43	9.10
FDKN [13]	0.69	5.78	173.01	5.37	7.97
DKN [13]	1.16	17.75	56.34	5.08	7.70
FDSR [9]	0.60	5.05	198.02	5.49	7.29
DuCos [46]	34.38	25.09	39.86	3.68	7.37
DegBins (RG [54])	16.28	9.75	102.6	3.63	7.18
DegBins (DuCos)	42.15	34.64	28.87	3.52	7.11

Tab. 6 presents a detailed complexity comparison on the real-world RGB-D-D dataset. DegBins first adopts DuCos [46] as the feature extraction backbone in Fig. 2, which incorporates the large-scale Depth Anything v2 [47]. While DegBins achieves the lowest error, it incurs higher computational cost and slower inference speed due to the substantial size of the depth foundation model. As an alternative, we replace the backbone with a lightweight residual group [54] for feature extraction. With only a marginal drop in accuracy, the number of parameters is reduced from 42.15M to 16.28M, and the inference speed improves by approximately 74 FPS. Moreover, DegBins-RG attains a lower RMSE than DuCos. These results suggest that exploring more efficient architectures with stronger accuracy–efficiency trade-offs is a promising direction for future work.

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 abstract and introduction clearly state our contributions and the claims made well match the experimental results.

   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 limitations of our work are discussed in Sec. 5.

   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: Our paper does not include theoretical results.

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: We provide the detailed implementation in the Sec. A.2.

   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: [No]

Justification: All datasets used in this study are publicly available.

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: We have provided the experimental details.

   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: [No]

   Justification: The datasets used in our experiments are large, and it is very costly to train our model for many times. In summary, we don't report the error bar.

   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: The computer resources have been provided in the experiment section.

   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: Our research conforms with the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: We have discussed the broader impacts in the main text.

    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: Our work poses no such risks.

    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: All assets used in our work are properly cited.

    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: [N/A]

    Justification: This paper does not release new assets.

    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: Our paper does not involve crowdsourcing nor 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: Our paper does not involve crowdsourcing nor research with human subjects.

    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: LLMs were not involved in any methodological component of the research.

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.