Perceptual Relationship and Representation Learning for 3D Understanding and Quality Enhancement

by

Xiaoyu Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024
© Xiaoyu Xu 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Jane Wang Professor, Dept. of Electrical and Computer Engineering, University of British Columbia
Supervisor(s):	Zhou Wang Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal Member:	Alfred Yu Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal Member:	En-Hui Yang Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal-External Member:	Alexander Wong Professor, Dept. of Systems Design Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Three-dimensional (3D) modeling plays a crucial role in a wide variety of real-world applications such as autonomous driving, smart cities, entertainment, education, and video game development. Over the past decades, research has focused on enhancing 3D understanding and improving the quality of 3D representation and rendering. Despite these efforts, existing approaches often rely on either low-level features, such as object sizes and edges, or high-level semantics, like object categories, while neglecting the critical relationships between multiple objects. These relationships are essential for a comprehensive understanding of 3D scenes in natural environments. Moreover, contemporary deep learning-based methods for 3D modeling often overfit due to reliance on large-scale neural networks, limiting their efficiency and generalizability. This thesis addresses these limitations through four key contributions centered on extracting efficient perceptual representations.

In the first part, we propose a method to extract perceptual relationships that extend 3D understanding from local cues, such as shapes and semantics, to global cues, such as depth perception. To achieve this, we design a Visual Relationship Spatialization (VRS) framework with two main objectives: (1) identifying perceptual relationships that significantly contribute to 3D understanding and (2) quantifying their contributions. By integrating these spatialized relationship representations into monocular depth estimation tasks, we evaluate their effectiveness. Experiments on KITTI, NYU v2, and ICL-NUIM datasets validate the efficacy of this approach. Furthermore, incorporating the relationship spatialization framework into state-of-the-art depth estimation models results in marginal improvements across most evaluation metrics.

In the second part, we extend perceptual relationship representations from single to multiple viewpoints. This extension enables the integration of richer 3D information into novel-view-related tasks, such as novel view synthesis (NVS), which requires generating multiple novel views. To this end, we introduce a visual relationship transformation (VRT) framework that predicts perceptual relationships from unseen viewpoints, thereby overcoming the constraints of view dependency. By capturing transformed relationship representations, this framework enhances 3D understanding in novel view synthesis tasks.

In the third part, we aim to improve the perceptual quality of rendered novel views, by

using human perceptual preferences (HPP), which are sensitive to distortions in 2D images. To this end, we design a human perceptual preference optimization (HuPPO) framework to improve the quality of 3D renderings. The framework imposes "human perception" as guidance to learn perceptually satisfactory representations. At the same time, the human perception is formulated into a meta-learning objective function to regularize the training process. Evaluation in the novel view synthesis task demonstrates the strong effectiveness of the proposed framework.

In the final section, we extend the aforementioned methods, which primarily enhance the quality of 3D modeling from perceptual representations, with a focus on improving efficiency and generalizability. To this end, we propose a hierarchical controllable diffusion model (HCDM) for learning 3D models, represented as neural radiance fields or point clouds, using a latent diffusion model. The designed HCDM improves the quality of 3D modeling by representing 3D models as functional parameters that are much fewer than the size of 3D models, allowing for an efficient representation. The HCDM is designed to learn the distribution of these parameters, improving generalizability across data from multiple modalities. The significant performance improvement of the proposed methods is further validated on 2D images and 3D motion data.

Acknowledgements

The journey to earning a PhD is not easy. I am deeply grateful to everyone who supported me through the difficulties. Without their help, I could not have overcome the challenges along the way.

I want to thank my advisor, Professor Zhou Wang, for his efforts in teaching me how to do good research. He is a role model to me with his rigorous academic style, insightful thoughts, and unwavering curiosity. His earnest dedication to research inspires and continually motivates me to progress.

I am also deeply grateful to my good friends, Sheyang Tang, Zhongling Wang, Jinghan Zhou, and Wenbo Yang, for accompanying me through life's challenges and sharing moments of joy along the way. I would also like to express my gratitude to my groupmates, Armin Shafiee Sarvestani, Mahdi Naseri, and Mahzar Eisapour, for their support in both life and research. I am sincerely grateful to Jiayan Qiu, my mentor of research, for providing kind and detailed guidance throughout my journey.

Finally, I am deeply thankful to my family—my mom, dad, sister, and girlfriend—for their mental support during challenging times and for bringing happiness and joy into my life.

Dedication

This is dedicated to the one I love.

Table of Contents

Examining Committee ii

Author's Declaration iii

Abstract iv

Acknowledgements vi

Dedication vii

List of Figures xiv

List of Tables xxii

List of Abbreviations xxv

1 Introduction 1

1.1 Motivation 3

1.1.1 Perceptual Relationships for 3D Understanding 3

1.1.2 Perceptual Relationship Transformation for 3D Understanding . . . 4

1.1.3 Human-perceptual-preferences for 3D Understanding 5

1.1.4 Efficient and General 3D Representations . . . 6
1.2 Objectives . . . 7
1.3 Contributions . . . 8
1.3.1 Visual Relationship Spatialization . . . 8
1.3.2 Visual Relationship Transformation . . . 9
1.3.3 Human Perceptual Preference Optimization . . . 9
1.3.4 Hierarchical Controllable Diffusion Model . . . 10
1.4 Publications . . . 10
1.5 Thesis Outline . . . 11

2 Literature Review **13**
2.1 Graph Neural Network . . . 13
2.1.1 Spectral-based GNN . . . 14
2.1.2 Methods and applications of spectral-based GNN . . . 17
2.1.3 Spatial-based GNN . . . 18
2.2 Relationship Recognition . . . 19
2.2.1 Two-stage methods . . . 19
2.2.2 One-stage methods . . . 23
2.2.3 Datasets for scene graph generation . . . 23
2.2.4 Applications of scene graph . . . 24
2.2.5 Methods statistics . . . 25
2.3 Depth Estimation . . . 25
2.3.1 Stereo-based depth estimation . . . 26
2.3.2 Monocular depth estimation . . . 27

2.3.3 Datasets for Stereo-based depth estimation . . . 31
2.4 Pedestrian trajectory prediction . . . 31
2.4.1 RNN-based methods . . . 32
2.4.2 CNN-based methods . . . 33
2.4.3 GAN-based methods . . . 33
2.4.4 Methods comparison . . . 34
2.4.5 Datasets for the PTP . . . 34
2.5 3D Modeling . . . 35
2.5.1 3D neural rendering . . . 35
2.5.2 Applications . . . 37
2.5.3 Generative 3D modeling . . . 38
2.5.4 Datasets for Generative 3D methods . . . 40
2.6 Implicit Neural Representation . . . 41
2.6.1 Implicit Neural Representation . . . 41
2.6.2 Generalized Implicit Neural Representation . . . 41
2.6.3 Generative Implicit Neural Representation . . . 42

3 Perceptual Relationships for 3D Understanding: 2D-image based monocular depth estimation 44

3.1 Introduction . . . 44
3.2 Method . . . 47
3.2.1 Stage 1: Representation Extraction . . . 47
3.2.2 Stage 2: Representation Modification . . . 49
3.2.3 Stage 3: Depth Estimation and Contribution Quantification . . . 50
3.3 Experiments . . . 51

3.3.1 Datasets and Implementation details . . . 52
3.3.2 Performance Evaluation . . . 53
3.3.3 Effect of Relationship Representation . . . 56
3.3.4 Effective Spatialized-Relationship Identification . . . 60
3.3.5 Respective Contributions Quantification . . . 63
3.3.6 Ablation Study . . . 64
3.3.7 Visualized results . . . 64

4 Perceptual Relationship Transformation for 3D Understanding: 2D-images based novel view synthesis 69

4.1 Introduction . . . 69
4.2 Method . . . 73
4.2.1 Stage 1: VRT . . . 74
4.2.2 Stage 2: Correspondence Search . . . 76
4.2.3 Stage 3: Optimization . . . 77
4.3 Experiments . . . 78
4.3.1 Datasets and Implementation Details . . . 78
4.3.2 Visual Relationship Transformation . . . 80
4.3.3 Ablation Studies . . . 81
4.3.4 Novel View Synthesis . . . 83
4.3.5 Pedestrian Intention Estimation . . . 86
4.3.6 Visual Grounding . . . 88
4.3.7 Results on KITTI . . . 89
4.4 Implementation details of downstream tasks . . . 91
4.4.1 Synsin [298] . . . 91

4.4.2 PixelNeRF [333] . . . 91
4.4.3 PIE [227] . . . 92
4.4.4 Visual grounding . . . 93

5 Human-perceptual-preferences for novel view synthesis with sparse views 95
5.1 Introduction . . . 95
5.2 Background knowledge . . . 99
5.2.1 Volume Rendering . . . 99
5.2.2 RLHF & DPO . . . 100
5.3 Method . . . 102
5.3.1 Candidate Selection . . . 103
5.3.2 Human Perceptual Preference Optimization . . . 105
5.3.3 Training scheme . . . 107
5.4 Experiments . . . 109
5.4.1 Datasets and Implementations . . . 109
5.4.2 Comparisons . . . 110
5.4.3 Distortions removal . . . 115
5.4.4 User study . . . 116
5.4.5 View-transformation robustness . . . 118
5.4.6 Ablation study . . . 119

6 Efficient and General 3D Representations for 3D Data Generation 121
6.1 Introduction . . . 121
6.2 Background knowledge . . . 124
6.2.1 Implicit Neural Representation and Generative INRs . . . 124

6.2.2 Hierarchy Analysis of INR . . . 125
6.3 Proposed Method . . . 126
6.3.1 Problem Statement . . . 127
6.3.2 Stage-1: Learning a dataset of LoE INRs . . . 128
6.3.3 Stage-2: Conditional distribution learning for generalizable INRs . . . 128
6.4 Experiments . . . 131
6.4.1 Details on Experimental Setup . . . 131
6.4.2 Quantitative and Qualitative Results . . . 132
6.4.3 Hierarchical Controllable Generation . . . 133
6.4.4 Analysis . . . 135
6.4.5 Ablation Studies . . . 138
6.5 Additional Experimental Results . . . 140
6.5.1 Generalizability Analysis Through Retrieval . . . 140
6.5.2 More Generated Samples . . . 140
6.5.3 AMASS Experiments . . . 140
6.5.4 Hierarchical Controllable Generation . . . 143
6.5.5 Latent-based Retrieval . . . 144
6.6 Additional Analysis . . . 146
6.6.1 Latent Space Analysis . . . 146
6.6.2 Binary condition Analysis . . . 147

7 Conclusion and Future Work **150**
7.1 Conclusion . . . 150
7.2 Future work . . . 152

References **155**

List of Figures

1.1 3D data formats: Point clouds, Mesh, and Neural Raidance Fields. 2

1.2 The illustration of how visual perceptual relationships help enhance 3D understanding. The left figure illustrates the single-view scenario, where an observer captures a relationship "behind" to enhance understanding of instance positions. The right figure illustrates the multi-view scenario, where perceptual relationship prediction from "behind" to "left to" across views helps reason instance positions in the novel view. 3

1.3 The illustration of utilizing human perception and efficient representations to improve the quality of 3D models. The left figure presents that the 3D models suffer from low perceptual quality in novel views. The right figure presents that 3D models can be represented with a compact and general neural network. 5

1.4 Thesis roadmap for solving monocular depth estimation (MDE), novel view synthesis (NVS), and 3D data generation (3D-gen). 12

3.1 Given a scene image (a), relationships contribute to depth estimation, as shown left to right in (b-d). Compared with the explicit spatial information from *in front of* (a), the implicit spatial information from *under* (b) also benefits the depth estimation. More interestingly, *wear* (c), which is intuitively non-spatial, in reality, contributes significantly to depth estimation. 45

3.2 Depth error decreases during our relationship spatialization process in the red boxed Table area (a-d). Only the visual understanding is utilized in (a); all Table-related in-image relationships are added in (b); only the effective relationships are identified and then equally added in (c); the identified relationships are contribution-accordingly weighted and then added(d). 46

3.3 The framework of the proposed model. An input image is firstly sent to an image feature extractor to generate image representations. The input image is also fed to relationship recognition module to generate relationship representations. then, the relationship ones are aligned to the image ones and being eliminated with redundancy. Finally, the effective relationship representations are quantified and concatenated with image features to estimate depth. 48

3.4 Image representation extractor 49

3.5 Representation modification is composed of spatial alignment and redundancy elimination. Firstly, relationship representations are spatially aligned towards image representations with bounding boxes. Secondly, the aligned relationship are orthogonalized against image representations to eliminate information redundancy. 51

3.6 Distribution of depth errors on relationship-related areas and non-related areas for NYUv2 dataset. The yellow boxes denote the *abs_rel* on the relationship-related areas, the blue boxes denote the *abs_rel* on the non-related areas, the in-box red line denotes median value, and the star symbol denotes the mean value. 58

3.7 Distribution of depth errors on relationship-related areas and non-related areas for KiTTI dataset. 59

3.8 Framework derivation visualization. The first column of each group shows the input scene image, the second column shows the framework derivation map, the third column shows the relationship representation map, the fourth column shows the depth error map, and the bounding box denotes the relationship area. 60

3.9 In each image, one target object of the identified effective relationships is erased. Then, users' judgments on whether the erased object is effective in depth estimation are recorded. 63

3.10 Quantification of relationships effectiveness. (a): ground truth image; (b)-(d): In the first round, erase each object and select the most important relationship; (e)-(g): In the second round, use image with the selected object erased as ground truth and select the next most important object. 63

3.11 Correlation between the depth error and its quantified respective contribution on relationship areas. 64

3.12 Visual examples on KITTI [82] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps. 65

3.13 Visual examples on NYU Depth v2 [256] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps. 66

3.14 Visual examples on ICL_NUIM [101] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps. 67

3.15 Visual examples on visual genome [132]. 68

4.1 Visual Relationship Transformation. Given an observed view image (a), we aim to predict the relationships between objects (e.g., the one between `car-1` and `car-2`) in novel views (b-d) without requiring their corresponding view images. Compared with the relationship observed in (a), the corresponding ones in novel views (b-d) are typically view-dependent and non-deterministic, which might be preserved, altered, or even become invisible. The faded images and relationships denote unseen and invisible, respectively. 70

4.2 Illustration of VRT implementations on Visual Grounding, Novel-View Synthesis, and Pedestrian Intention Estimation tasks. For each task, the first image denotes the given input, the second one denotes the result from the SOTA method, the third one denotes the result by adopting our VRT as a plug-in module, and the fourth one denotes the ground-truth. 72

4.3 The illustration of the proposed approach. The T_{st} denotes the transformation matrix from the observed source view to the target view. $\bigotimes$ denotes the re-indexing and alignment operation. Note that, the relationship values and presentness are predicted by the VRT module, and the structure information is constructed by the predicted relationship and presentness. 73

4.4 Visual results from Synsin [298] on Realestate10K [370]. The first two columns show the images from the original view and the target view, the third one shows the synthesized images from Synsin, and the fourth one shows the images from Synsin+ that adopt VRT as a plug-in module. The right four columns present that the synthesized images get better quality when relationships are predicted correctly. The dashed boxes denote invisible relationships in the target views. 84

4.5 Visual results from Synsin [298] on Realestate10K [370]. The first column shows the input image, the second and the fifth columns show the output of enhanced and original Synsin, and the third and the fourth columns present the zoomed-in parts.. 85

4.6 Visual results from PixelNeRF [333] on DTU MVS [1]. 86

4.7 Visual results from RerferIt3D [3] and RerferIt3D+. 88

4.8 Framework of Synsin [298] 92

4.9 Framework of Synsin+. 92

4.10 Framework of PixelNeRF [333] 92

4.11 Framework of PixelNeRF+. 92

4.12 Framework of PIE+. 93

4.13 Framework of ReferIt3D [3] 94
4.14 Framework of ReferIt3D+. 94

5.1 **Artifacts illustration**. Each bar plot presents the quality of renderings versus *training steps* and *viewpoints*. Our method addresses quality fluctuation (the green dot), distant view degradation (the yellow dot), and geometry inconsistency (the red dot). The left column of images is rendered by SPARF[282], and our method produces more perceptually satisfactory results as shown in the right column. 96
5.2 Human perception guides the parameters of the neural radiance fields (NeRF) model (a) to progressively converge toward a perceptually satisfactory direction (c), which is better than a merely meta-learned model (b). 98
5.3 Human perception enhances the model's resilience against noises on view transformation. (b) shows that the human perception-guided field representations have fewer errors than vanilla NeRF (a). 99
5.4 The proposed framework includes three modules: (1). candidate selection creates positive and negative renderings. The positive rendering is generated from a hyper-restoration module. The negative rendering is generated from perceptually adversarial attack; (2). HuPPO objective is constructed on the positive and negative renderings; (3). Meta-training first optimizes original supervisions in the I-step, then utilizes positive, negative, and supervised renderings to optimize the HuPPO objective in the H-step, and finally learns the general model in the M-step. 102
5.5 Qualitative comparisons of state-of-the-art (SOTA) methods NeRF [333] (first row), Depth-NeRF [49] (second row), and Sparf [282] (third row). The red-boxed figures present the original models' results while the yellow-boxed figures present the enhanced results with HuPPO optimization. 113

5.6 Geometry rendered by marching cubes [172]. The first and the third columns plot two different views of the mesh learned by NeRF [192] on 3 supervision views. The second and fourth columns present two views of the mesh learned by the NeRF enhanced with the HuPPO. 114

5.7 Distant view degradation. The left plot illustrates which views we select. The green view camera represents the supervision view, and the brown and red cameras denote novel views. The top 3 renderings are from the sparf [282] trained on 3 supervision views, the bottom 3 are enhanced with HuPPO. 116

5.8 Distant view degradation. The perceptual quality of distant views of Enhanced Sparf is better than that of Sparf and DsNeRF. 117

5.9 Quality fluctuation: the peak signal noise ratio (PSNR) v.s. training stages of the two nearest neighbors to the supervised views are denoted as "NeRF1" and "NeRF2". The dynamics of HuPPO are denoted as "Our1" and "Our2". . . . 117

5.10 Ablation study on the performance of different NR-IQA models. 120

6.1 Hierarchical generation of universal data modality. *Top*: expanded representational capabilities of a Layers-of-Experts implicit neural representation (INR) model. *Bottom*: aligning these capabilities with hierarchical data patterns enables precise control over the generation process. Each column presents generated samples resulting from a divergence in routing at a specific layer, with arrows indicating the shared routing in the preceding layers. 122

6.2 The proposed Controllable Hierarchical Implicit Neural Representation (CHINR) consists of two stages. In Stage-1, a Layer-of-Experts (LoE) model is used to represent large-scale data, utilizing instance-specific latents and shared experts. Stage-2 introduces a Hierarchical Controllable Diffusion Model to learn the layer-wise conditional distributions of the latents. 126

6.3 Condition formation. Layers of latent $\mathbf{h}$ are first concatenated and then compressed. The compressed tensors are binarized to generate low-dimensional binary conditions. 129

6.4 Uncurated generations for ShapeNet, CelebA-HQ, and SRN-Cars datasets. 133

6.5 Controllable hierarchical generation by progressively fixing the layer-wise latent. 133

6.6 Latent composition. The first row presents 9 randomly sampled images from Stage-2. The second to the fourth rows present the images where the second to the fourth parts of the first sample's latent (green boxes) are replaced. The representative examples are highlighted in red boxes. 136

6.7 Layer-wise hierarchy analysis. (a): correlation between cross-layer latents, obtained on the train-split of CelebA-HQ at Stage-1. (b) and (c): visualization of conditional distributions across layers. The gray regions show the distribution of latents from Stage-1, while the colored regions represent the sampled latents from Stage-2. 138

6.8 Ablations on conditional modeling. 139

6.9 Retrieval on CelebA-HQ: mNIF retrieves images closely resembling those from the training set, while CHINR demonstrates better generalization by producing distinct new images. 141

6.10 More generated samples of CelebA-HQ, ShapeNet, and SRN-Cars data. . . 142

6.11 Generated motions with HCDM. each row denotes one sampled data. . . . 142

6.12 Semantic interpolation for AMASS data. Anchor sequential motions (indicated by the red and green dashed boxes) are first fitted with layer of experts (LOE) to obtain latents. Then semantic-level interpolation is performed by interpolating the latents. The red dashed box denotes the start motion, and the green dashed box denotes the end motion. 143

6.13 More examples of hierarchical controllable generation on CelebA-HQ, ShapeNet, and SRN-Cars data. 144

6.14 Latent-based retrieval via two approaches: retrieval by all layers and retrieval by each layer. 145

6.15 Latent space interpolation is performed for LoE, with four corner points representing the anchor examples rendered in stage 1. The intermediary points are generated through the bilinear interpolation of the latents associated with these four anchors. The interpolation is evaluated on datasets CelebA-HQ, ShapeNet, and SRN-Cars. 146

6.16 Layerwise interpolation. The red boxes denote the start and the green boxes denote the end. For the CelebA-HQ, the layers $2 \rightarrow 4$ are interpolated respectively while other layers are fixed. For the ShapeNet, the layers $1 \rightarrow 3$ are interpolated respectively. 147

6.17 Correlation between the learned latents across layers, trained on ShapeNet [23] and SRN-Cars [259]. The non-negligible correlation between adjacent layers (e.g., $\mathbf{h}^1$ and $\mathbf{h}^2$) reveals the necessity of conditional distribution learning. . 148

6.18 Visualization of conditional distributions across layers $3, 4, 5$. The gray regions present the distribution of latents from Stage-1, while the colored regions represent the sampled latents from Stage-2. 148

6.19 Clusters of each part of latent and binary conditions. The dotted plot presents clusters of each part of latents trained on ClebA-HQ. The gray distribution plot presents the distribution of each part of latents, and starred scatter plot presents clusters of latents with similar binary conditions. . . 149

List of Tables

2.1 Statistics of some state-of-the-art methods. 25

2.2 Datasets for depth estimation. 32

2.3 State-of-the-art methods for the pedestrian trajectory prediction. 34

2.4 Methods and Evaluation of 3D generative methods. ↑ denotes the higher the better, while ↓ denotes the lower the better. 39

2.5 Datasets for 3D data generation. 40

3.1 Results of monocular depth estimation on KITTI dataset. ↑: the higher is better. ↓: the lower is better. 54

3.2 Results of monocular depth estimation on NYU Depth v2 dataset. 54

3.3 Results of frameworks trained on NYU Depth v2 dataset and evaluated on ICL_NUIM dataset. 55

3.4 The model added with relationship representations (+R) outperforms baseline model only using image representations (B) both on KITTI and NYU datasets. 57

3.5 The monocular depth estimation (MDE) performance increases with the relationship recognition accuracy. *Recall* is used as the evaluation metric of the pre-trained relationship recognition module, $User_Acc$ is used as the metric for user-oriented accuracy and $rmse$ is used as the metric for MDE. 57

3.6 The model added with effective relationship representations (+ER) outperforms the model with equally treated relationship representations (+R) both on KITTI and NYU datasets. 61

3.7 The model added with quantified relationship representations (+QR) outperforms the model only using effective relationship representations (+ER) both on KITTI and NYU datasets. 61

3.8 The model added with redundancy elimination (+RE) outperforms the baseline model only using image representations (B) both on KITTI and NYU datasets. 61

3.9 Quantification of relationship effectiveness. 62

4.1 Results evaluated with Recall@50 and Recall@100 on tasks: relationship classification (PredCls) and scene graph classification (SGCls). 80

4.2 Results compared with VRDFormer [367]. 81

4.3 Results of VRT with different transformations. 82

4.4 Results of involving the transformation information in different stages of information aggregation. 82

4.5 Results of ablation study. 82

4.6 Results on Realestate10K [370] dataset. Synsin+ denotes enhanced with VRT . 84

4.7 Quantitative results on different transformation scales. 87

4.8 Results on DTU MVS [1]. 1v and 3v denote the number of available views in the training process. 87

4.9 Results on Nr3d [3] datasets. 87

4.10 Results on PIE [227]. 87

4.11 Results evaluated with Recall@50 and Recall@100. 89

4.12 Results and ablation study on KITTI. 90

4.13 Results of VRT with different transformations. 90

4.14 Results of involving the transformation information in different stages of information aggregation. 91

5.1 Comparison with SOTA methods on DTU. We retrain and evaluate the SOTA methods on 3, 6, and 9 viewpoints and report the performances of PSNR, SSIM, LPIPS, and non-reference image quality assessment (NR-IQA). The "Type" column exhibits the features or regularizations of the methods. The light gray rows present methods enhanced with the HuPPO framework, e.g. Nerf(h). the ↑ denotes the higher the better and the ↓ denotes the lower the better. 111

5.2 Comparison with SOTA methods on LLFF. 112

5.3 Comparison with SOTA methods on Blender. 112

5.4 Subjective test for perceptual quality assessment of 5 scenes. 118

5.5 Ablation study of results under different view perturbation. 119

5.6 Ablation study of results for different candidate views. 119

6.1 Quantitative results on CelebA-HQ, ShapeNet, and SRN-Cars. 134

6.2 Ablation study of different binary condition lengths, 8, 12, 15, and 20, when training Stage-2 on CelebA-HQ, ShapeNet, and SRN-Cars datasets. 134

6.3 Quantitative results on AMASS. 140

List of Abbreviations

3D Three-dimensional iv

CHINR Controllable Hierarchical Implicit Neural Representation xix, xx, 126, 132, 140, 141

CNN convolutional neural network 3, 21, 25–29, 32, 33, 37, 86, 91

CRF conditional random fields 28, 30

DPO direct preference optimization 100

EMP Equivariant Message Passing 74

GCN graph convolutional network 14, 18, 29–31

GNN graph neural network 13–19, 22, 81

HCDM hierarchical controllable diffusion model v, xx, 7, 8, 10, 124, 129, 130, 132, 133, 135–137, 139, 142, 147, 151

HPP human perceptual preferences v, 6, 9

HuPPO human perceptual preference optimization v, xviii, xix, xxiv, 6, 8–10, 98, 99, 102, 103, 105–108, 110, 111, 113–118, 151

INR implicit neural representation xix, 41–43, 121–128, 131, 137, 151

IQA image quality assessment 98, 151

LIDAR Light Detection and Ranging 1, 2

LOE layer of experts xx, 123, 127, 128, 131, 132, 135, 136, 140, 142–144, 147, 151, 152

LSTM long short-term memory 19, 32

MDE monocular depth estimation xxii, 4, 11, 27, 44, 45, 47, 51–53, 56–60

MLP multi-layer perceptron 1, 7, 37, 38, 41, 76

MOE mixture of experts 123

MOS mean opinion score 97

NeRF neural radiance fields xviii, xix, 37, 38, 95–100, 102–110, 114–116, 118, 119, 151, 152

NR-IQA non-reference image quality assessment xxiv, 98, 103, 105, 106, 109–111, 116, 118, 119

NVS novel view synthesis iv, 11, 69, 70, 151

PIE pedestrian intention estimation 31, 32, 72, 78, 92

PSNR peak signal noise ratio xix, xxiv, 110, 111, 114, 115, 117, 119

PTP pedestrian trajectory prediction 31–34

QoE quality of experience 6

RLHF reinforcement learning with human feedback 100, 101, 106

RNN recurrent neural network 32

ROI region of interest 21

SGG scene graph generation 19–21

SOTA state-of-the-art xviii, xxiv, 109–114, 151, 152

TCN temporal convolutional networks 30, 31

TOF time-of-flight 1, 2

VN-MLP Vector-Neuron Multi-layer Perceptron 75

VRD visual relationship detection 70, 80, 81, 89

VRS Visual Relationship Spatialization iv, 4, 7, 8, 150

VRT visual relationship transformation iv, xvii, xxiii, 5, 8, 9, 70–73, 76–84, 86, 88–92, 151

VT-robust view transformation-robust 97, 98

Chapter 1

Introduction

3D modeling involves creating a coordinate-based representation of an object's surface in three dimensions by manipulating edges, vertices, and polygons within a simulated 3D space. As a prominent topic in computer graphics, it has spurred significant research interest in areas such as novel view synthesis, point clouds, and volume rendering. 3D data can be represented in various formats, as shown in Fig. 1.1. Generally, a 3D data instance is composed of appearance and geometry. For point cloud data, the appearance is represented by the RGB values, and the geometry is represented with the 3D coordinates. For mesh data, the appearance and the geometry are represented with texture maps and vertices respectively. For the neural radiance fields, they adopt a multi-layer perceptron (MLP) to represent appearance as RGB values, and geometry as the density of each point in the field. To make 3D data applicable to different tasks, quality, and efficiency are critical factors determining its practical applicability across various domains.

However, existing techniques to achieve higher-quality 3D models often rely on sophisticated and expensive equipment, such as Light Detection and Ranging (LIDAR) systems or time-of-flight (TOF) cameras, to capture real-world scenes. However, the demand for greater flexibility and cost-efficiency has shifted research focus toward generating 3D models from multiple-view or single-view 2D images. Compared to LIDAR and TOF cameras, 2D images offer several significant advantages: *Flexibility*: 2D images can be easily captured at any time and place using a simple camera, offering flexibility in terms of time,

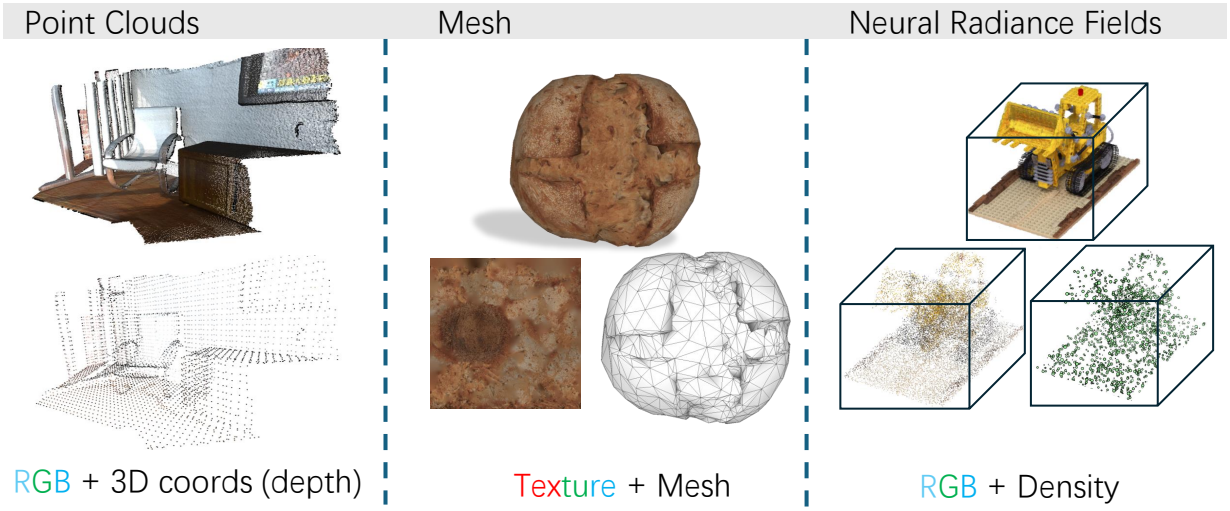


Figure 1.1: 3D data formats: Point clouds, Mesh, and Neural Raidance Fields.

location, and device. *Cost-effectiveness*: LIDAR and TOF systems are currently more expensive than standard cameras or smartphones, making 2D imaging a more economical alternative. *Storage efficiency*: Storing 3D data requires much more space than 2D images, as it includes depth information for each pixel, whereas 2D images are far less storage-intensive. Therefore, extracting 3D information from 2D images shows high potential and plays a crucial role in promoting 3D modeling techniques. Nevertheless, this is challenging, because 2D images lack most depth information, making direct 3D modeling difficult. So it is crucial to infer 3D structures through 2D visual understanding, such as perceptual relationships. To this end, this thesis focuses on learning rich 2D visual understandings and representations to enhance 3D understandings and quality. Besides, the learned representations are also stored in an efficient and general way. In the following sections, we outline the motivations for pursuing higher-quality and more efficient 3D modeling, followed by a chapter-by-chapter clarification of our objectives. We then detail our contributions compared to existing methods and highlight the associated publications. Finally, we provide a summary of the thesis roadmap.

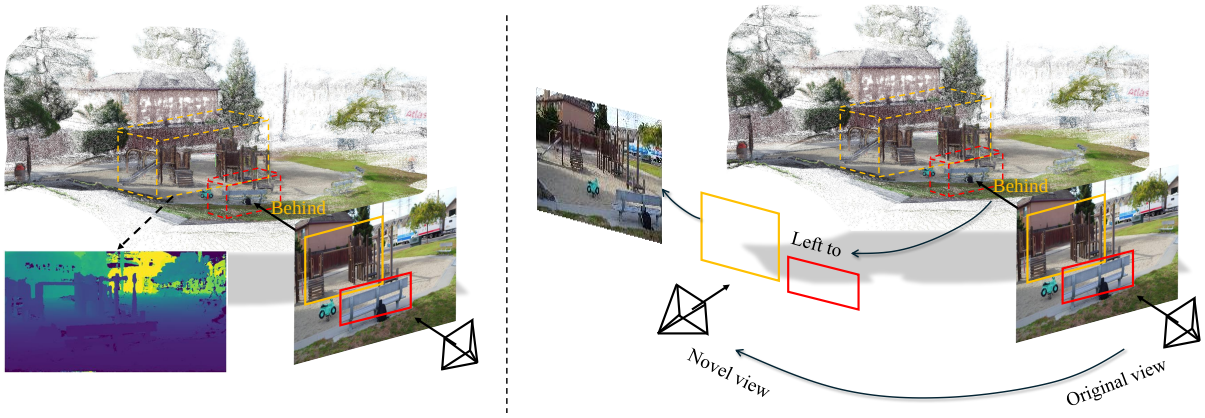


Figure 1.2: The illustration of how visual perceptual relationships help enhance 3D understanding. The left figure illustrates the single-view scenario, where an observer captures a relationship "behind" to enhance understanding of instance positions. The right figure illustrates the multi-view scenario, where perceptual relationship prediction from "behind" to "left to" across views helps reason instance positions in the novel view.

1.1 Motivation

1.1.1 Perceptual Relationships for 3D Understanding

In the natural scene, humans are able to estimate 3D information with monocular images, even without explicit 3D stimulation, i.e. binocular parallax. This is because we can reason the depth information of objects by their perceptual relationships. As shown in the left image of Figure 1.2, the object "playground structure" is "behind" the "bench". With the relationship "behind", it indicates the depth of the "playground structure" should be larger than the "bench". Here the larger the depth denotes the farther to the camera. However, current methods for estimating 3D information, i.e. depth, from 2D images typically rely on pixel-level relationships. These approaches use convolutional neural network (CNN)s to capture pixel correlations within specific receptive fields. Specifically, they employ an encoder-decoder architecture that extracts low-level information, e.g. shape, contour, and contrast of 2D images. However, low-level information alone offers only pixel-based cues, which overlook the high-level correlations, thus limiting the depth estimation's precision.

To further improve accuracy, high-level semantics, i.e. the perceptual relationships between objects in a scene offer valuable 3D insights. Recognizing a relationship like "Playground structure behind the bench" offers crucial instance-level depth cues.

To leverage depth cues in perceptual relationship representations to enhance 3D understanding, we try to figure out: (1) Which relationships are effective for depth estimation; (2) How to quantify their contributions? In Chapter 3, we design a VRS framework to answer the questions. Specifically, the relationship spatialization adopts a relationship detector to extract perceptual relationship representations. Then these representations are "spatialized" to align with the shape of input images. Further, the aligned relationship representations are taken as input to an attention module to identify and quantify effective ones. With the proposed framework, we can estimate more accurate depth from monocular images. Extensive experiments on the MDE task demonstrate the effectiveness of our method.

1.1.2 Perceptual Relationship Transformation for 3D Understanding

Estimating depth from monocular images is convenient when 3D models are not required. However, in many scenarios such as 3D movies, the developers always need to construct 3D models from multiple 2D images and use them to render novel views. In these cases, 3D understanding is crucial for the quality of constructed 3D models and rendered novel views. However, the perceptual relationship is not accessible when we want to render novel views. As shown in the right figure of Figure 1.2, if we know the perceptual relationship of "playground structure behind bench" will transform into the "playground structure left to bench" in the novel view, providing the transformed relationship representation helps get more accurate 3D information thus higher quality rendered images. In another aspect, perceptual relationships derived from a single image are highly view-dependent and can be ambiguous when applied to novel view synthesis. Directly involving them in novel view synthesis would mistakenly infer depth and spatial relationships from different angles. Therefore it is crucial to establish robust relationship understanding to effectively leverage the rich 3D information provided by perceptual relationships in novel view synthesis. This

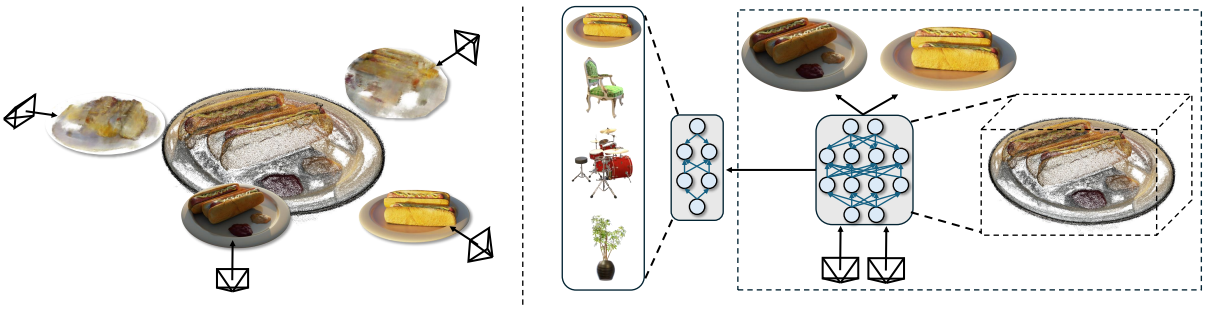


Figure 1.3: The illustration of utilizing human perception and efficient representations to improve the quality of 3D models. The left figure presents that the 3D models suffer from low perceptual quality in novel views. The right figure presents that 3D models can be represented with a compact and general neural network.

helps to avoid depth ambiguity and ensures accurate and consistent generation of new viewpoints.

In Chapter 4, we propose a VRT framework to dig out 3D information in perceptual relationships and impose them for novel view synthesis. Learning the transformed relationship is challenging in two aspects: (1) the variation of relationships between original and novel views can be large. For example, a relationship may disappear, emerge, or change in the novel view; (2) The predicted relationships should be structurally consistent with the original ones. To solve the challenges, we propose two techniques: (1) separate learning of relationships and presentness; and (2) graph structure matching. Extensive results on novel view synthesis demonstrate the effectiveness of the method. Moreover, we find that transformed relationships are also robust to large-view transformation. So they are applicable for many downstream tasks, such as 3D object referring, and pedestrian trajectory prediction.

1.1.3 Human-perceptual-preferences for 3D Understanding

Perceptual relationships provide valuable information for 3D understanding but are limited to scenarios involving multiple objects. Recently, there has been growing interest in representing single 3D objects, where such relationships do not apply. As illustrated on the left

side of Figure 1.3, rendering novel views of 3D models can result in low perceptual quality, often due to a limited number of available 2D views. In these cases, HPP can help generate more photorealistic rendering. A key advantage of human perception is its sensitivity to a wide range of visual and contextual factors beyond perceptual relationships. HPP, often quantified as the rank of quality of experience (QoE), is sensitive to both pixel-level and instance-level inconsistencies. For example, pixel-level distortions, such as anomalies in an object's edges or textures, can lead to lower QoE. At the instance level, irregularities in human faces or object shapes also detract from perceived quality. Furthermore, HPP evaluates the overall fidelity of 2D images, offering critical constraints to maintain global coherence and quality in 3D rendering. Human perception is more sensitive to distortions that might otherwise go unnoticed in perceptual relationships. For instance, noise in objects may not impact the recognition of visual relationships but can still be detected by human perception. This makes HPP a more sensitive and comprehensive source of information for improving 3D understanding and enhancing rendering quality.

To harness the power of human perception, we design a HuPPO framework, as presented in Chapter 5. Instead of using representations as the above methods, we formulate the human perception preference into loss function. Specifically, the proposed method contains three parts: (1) The candidate selection module. Since evaluating the perceptual preference on every view is infeasible, we instead count on the human perception to select reasonable candidates for enhancement; (2) HuPPO objective formulation. We formulate the human perception into HuPPO objective given the selected candidates. (3) Meta-training pipeline. Since meta-learning helps extract shared data features, we add an extra H-step to learn the perceptually satisfactory features shared between original supervision and anchor viewpoints. Experiments on few-shot neural radiance fields demonstrate the effectiveness of our framework.

1.1.4 Efficient and General 3D Representations

The above methods discuss how to leverage perceptual relationship representations to enhance 3D understanding and quality. In another aspect, representation efficiency and generalizability are also crucial. When storing 3D models as RGB and depth values, the

storage requirement amounts to at least $4 \times N \times H \times W$ parameters, where N, H, W represent the number of views, image height, and width, respectively. This storage consumption scales with the number of pixels, making it impractical for high-quality images. As an alternative, 3D data can be represented as a continuous function using a coordinate-based neural network where only the model parameters need to be stored. The storage requirement remains constant regardless of the number of pixels, offering a far more efficient solution than depth maps, meshes, or point clouds. As shown in the right of Figure 1.3, existing 3D models such as neural radiance fields utilize neural networks to overfit a scene with multiple 2D images. The overfitting brings about two limitations: 1. the size of the neural network is inevitably large to get better quality; 2. the generalizability is limited.

To this end, we explore the general representation of 3D data to balance representation efficiency and quality and propose a HCDM to learn the general representations in Chapter 6. Specifically, the proposed method contains two stages, for stage 1, each data instance is fitted with MLPs and collects parameters of all MLPs. In stage 2, the proposed HCDM is utilized to learn the distribution of collected parameters. The HCDM can learn general representations by modeling the hierarchical structures of the MLPs. We first evaluate the model on 2D images and then apply it to multiple data modalities such as point clouds, neural radiance fields, and 3D motion. The results show the proposed method can generate novel data on any data modality with higher diversity. Moreover, the hierarchical representations also allow for controllable data generation.

1.2 Objectives

This thesis targets enhancing the quality and efficiency of 3D modeling by leveraging perceptual relationships and representations from 2D images. The specific objectives are summarized as follows:

- In Chapter 3, we introduce VRS, a method designed to extract perceptual relationships in 2D images to enhance 3D understanding. With the extracted relationships, we apply them to monocular depth estimation.

- In Chapter 4, we present VRT, which aims to enhance the quality of synthesized novel views through transformed relationship representations. These transformed relationships capture spatial information in novel views where the original relationships are unavailable.
- In Chapter 5, we propose HuPPO, which aims to leverage human perception to help improve the quality of novel view synthesis. It is more applicable than perceptual relationships in single-object scenarios.
- In Chapter 6, we propose HCDM to learn efficient and general representations for 3D data generation. The hierarchical structure of representations also allows for controllable generation.

1.3 Contributions

This section outlines the key contributions of the thesis, providing a detailed summary of the contributions made by each project.

1.3.1 Visual Relationship Spatialization

The VRS framework, presented in Chapter 3, works on enhancing 3D understanding with extracted perceptual relationship representations. Its contributions are summarized as follows.

- We propose a novel framework that makes the first attempt to spatialize the relationships for depth estimation. The framework addresses the limitations of previous works, which rely solely on semantics, and often lack global constraints.
- We design a relationship spatialization module and apply an attention mechanism, with which the spatial priors from relationships are mined automatically, identified effectively, and quantified adaptively.

- Our proposed framework can be implemented on any deep learning-based monocular depth estimation method for performance boosting. Experiments on all datasets show encouraging and promising results.

1.3.2 Visual Relationship Transformation

The VRT framework, presented in Chapter 4, works on predicting transformed relationships for unseen views, the transformed relationship representation helps enhance 3D understanding. Its contributions are summarized as follows.

- We introduce a new task, termed VRT, along with a novel equivariant graph network designed to tackle this task. The proposed method predicts multi-view relationship understanding, offering global information for 3D modeling, in contrast to previous methods that rely solely on local semantic information.
- To evaluate the recognition accuracy of our VRT, we conducted extensive objective and subjective experiments and our VRT obtained encouraging results.
- Moreover, by adopting our VRT as a plug-in module, state-of-the-art methods in 3D visual grounding, novel-view synthesis, few-shot neural radiance fields, and pedestrian intention estimation have achieved considerable qualitative and quantitative performance improvement.

1.3.3 Human Perceptual Preference Optimization

The HuPPO relies on human perception preferences to improve the quality of 3D rendering, which also adopts a meta-training pipeline. The contributions are summarized as follows.

- This work presents the first framework to incorporate HPP into direct preference optimization, termed HuPPO, enabling the use of perception scores as the objective function. The introduced HPP significantly enhances the perceptual quality of rendering, a factor that has been largely overlooked within the 3D modeling community.

- We theoretically define a HuPPO loss function that leverages human perception as guidance to optimize training perspectives. The loss function facilitates the learning of perceptually satisfactory and robust features.
- We design a positive and negative candidate generation module. It includes a hyper-restoration module to increase novel views in the condition of ground-truths shortage. The module can alleviate different levels of degradation by exploiting human perception scores as hyper-modulation.

1.3.4 Hierarchical Controllable Diffusion Model

The HCDM adopts a two-stage pipeline to achieve efficient and general representations for 3D data. Its contributions are summarized as follows.

- We model 3D data representations using a Layer-of-Experts framework to align the inherent hierarchy of data semantics with the model architecture, the alignment enables fine-grain control of data generation. In contrast, existing methods overlook the hierarchical structure, resulting in a lack of generalizability and control over data generation.
- We propose modeling conditional dependencies within parameters using the proposed HCDM, enabling hierarchical control over the generated data semantics at varying levels of granularity.
- We offer theoretical analysis and empirical evidence to show the inherent connection between the hierarchy of parameters and data semantics, validated through extensive experiments across various modalities.

1.4 Publications

While studying the project, we achieved significant progress, which has been published or submitted to conferences and journals. We provide a list of our papers and their connections to the chapters as follows:

- The paper titled "Relationship spatialization for depth estimation" was published in the 16th European Conference on Computer Vision (ECCV) 2022, and contributes to Chapter 3.

- The paper titled "Visual Relationship Transformation" was published in the 18th European Conference on Computer Vision (ECCV) 2024, and contributes to Chapter 4.

- The paper titled "Human Perception-Guided Meta-Training for Few-Shot NeRF" was published in the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2024, and contributes to Chapter 5.

- The paper titled 'Controllable Data Generation with Hierarchical Neural Representations" was submitted to the Thirteenth International Conference on Learning Representations (ICLR) 2025, and contributes to Chapter 6.

1.5 Thesis Outline

With the objectives and contributions, the thesis roadmap is summarized in Fig. 1.4. We extract perceptual relationship representations to solve MDE and NVS as presented in Chapter 3 and 4. Then, the perceptual preferences are extracted to enhance 3D modeling quality as presented in Chapter 5. Finally, we construct efficient and generalizable representations for 3D data generation as presented in Chapter 6.

The structure of the following thesis is organized as shown in follows:

- Chapter 2 provides a comprehensive background study and literature review of graph neural networks, relationship recognition, depth estimation, pedestrian trajectory estimation, 3D modeling, and implicit neural representation.

- Chapter 3 describes the methodology, implementation, and experiment of the proposed "Visual relationship spatialization". It also includes extensive analysis and ablation studies to evaluate the effectiveness of the proposed method.

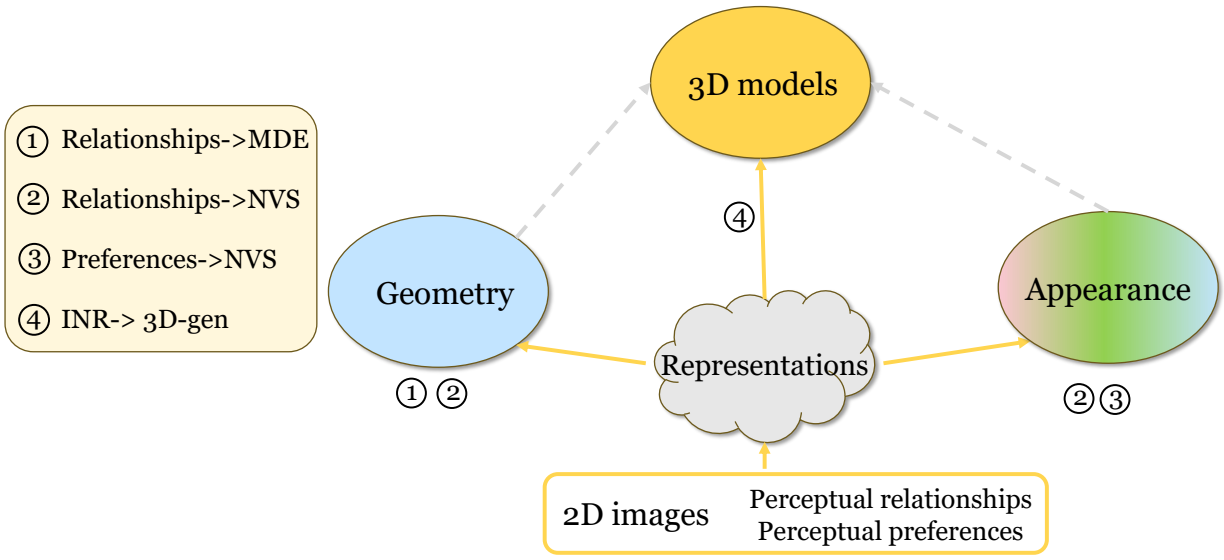


Figure 1.4: Thesis roadmap for solving monocular depth estimation (MDE), novel view synthesis (NVS), and 3D data generation (3D-gen).

- Chapter 4 explains the detailed method and implementation of the proposed "Visual relationship transformation" framework. It also presents the potential applications the framework can work on.
- Chapter 5 provides a formulation of "human perceptual preference optimization" and proposes a meta-training framework to enhance 3D quality.
- Chapter 6 explores the efficiency and generalizability of 3D representation.
- Finally, Chapter 7 concludes the thesis and presents a promising outlook for future research.

Chapter 2

Literature Review

In this chapter, we review the prior works related to our thesis, including graph neural networks, relationship recognition, depth estimation, pedestrian trajectory prediction, 3D modeling, and implicit neural representation.

2.1 Graph Neural Network

A graph neural network (GNN)[217, 136, 180, 147, 286, 109, 26, 11, 78, 197, 98, 88, 204, 10, 2, 140] is a deep learning model specifically designed to process data represented as graphs—mathematical structures comprising nodes (vertices) and edges that model pairwise relationships between entities. GNNs excel at capturing the relationships and dependencies between nodes, making them particularly effective for tasks involving non-Euclidean data, such as social networks, molecular structures, and knowledge graphs. Earlier works on GNN often relied on pre-defined node and edge features or aggregated these features iteratively[293, 294, 179, 218], a process that is both computationally expensive and time-consuming [76, 242, 93, 262]. More recently, approaches have been proposed to learn these features directly, streamlining the process and improving efficiency. Specifically, the spectral-based GNN implement the spectral theory for graph convolution design [143, 127, 44, 102, 17], and spatial-based ones utilize the mutual dependency by designing

the information aggregation manner [169, 324, 322, 321, 352, 285, 220, 219, 339, 233, 163, 335, 336, 288, 325]. More recently, the hierarchical graph convolutional network (GCN) [330] with high representation capability is proposed and shows promising performance. In the following parts, I will introduce the theory, applications, and other details of the graph neural network.

2.1.1 Spectral-based GNN

The spectral-based GNN was first proposed by Shuman [255] to process irregular high-dimension data based on the spectral graph theory, which involves with convolving input signal f with a kernel H. However, applying standard convolution operations to irregular data presents several challenges:

1. A "classical" signal $f(t)$ has a concept of "translate to the right by 3" to get $f(t-3)$. But for graph signal, it is not clear to say "translate by 3". Because it doesn't have a clear order, thus is not "shift-invariant".

2. Multiplying a signal with a complex exponential signal corresponds to translation in the Fourier domain. However, the signal on the graph is irregularly spaced, making it difficult to define translation in the spectral domain.

3. It's difficult to down-sample a signal on graph vertices.

4. There is no easy method to create a coarser graph that somehow captures the structural properties of the original one.

5. There is no easy localized operation to compute information from small neighboring vertices.

Problem settings

To solve the challenges, the spectral graph is proposed. The problem settings of the spectral graph theory are explained as follows. Assume we have an undirected weighted graph G, which contains vertices V with $|V| = N$ (each vertex has a C-dimensional feature vector), a set of edges E, and an adjacency matrix W. If the edge $e(i, j)$ which connects two vertices i, j exists, the entry $w_{i,j}$ denotes the weight of the edge; otherwise $w_{i,j} = 0$.

A signal defined on a graph is denoted as $\{f : V \to \mathbb{R}\}$. If the graph has N vertices, then $f = [f(v_1), \cdots, f(v_N)]$. In GNN, f is generally represented by embedded features with $f \in \mathbb{R}^{C \times N}$, where C denotes the shape of features.

The Graph Laplacian

The Graph Laplacian is at the core of the spectral graph theory. Here the Graph Laplacian refers to the unnormalized one. Generally in a graph, the Laplacian is defined as $L = D - W$, where D is a degree matrix, which is diagonal with the i-th diagonal entry $D_i = \sum_j w_{i,j}$. Given the defined signal f on the graph, the Laplacian operation of the f is formulated as:

$$(Lf)(i) = \sum_{j \in N_i} w_{i,j}[f(i) - f(j)]$$

.

The Laplacian operator is a difference operator, describing the average difference level between the vertices i and its neighboring vertices N_i.

There are some definitions and properties for the Laplacian matrix:

1. The Laplacian matrix is a real symmetric matrix, it has a complete set of orthonormal eigenvectors, denoted as $\{u_l\}$, and corresponding real non-positive eigenvalues $\{\lambda_l\}_{l=0,1,\cdots,N-1}$

2. The number of zero eigenvalues indicates the number of connected components, namely sub-graphs.

3. The eigenvalues of Laplacian matrix is sorted as $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \cdots \leq \lambda_{N-1} = \lambda_{max}$. The spectrum of the Laplacian matrix is defined as $\{\lambda_0, \lambda_1, \cdots, \lambda_{N-1}\}$

The graph transform

Recall the classical Fourier transform on a one-dimensional continuous signal

$$\hat{f}(\xi) =< f, e^{2\pi j \xi t} >= \int f(t) e^{-2\pi j \xi t} dt$$

.

The Fourier transform can be understood as the expansion of the signal $f(t)$ with the complex exponentials $e^{2\pi j\xi t}$. What is interesting is the complex exponential $e^{2\pi j\xi t}$ is an eigenfunction of the Laplacian operator

$$-\Delta(e^{2\pi j\xi t}) = -\frac{\partial^2}{\partial t^2}e^{2\pi j\xi t} = K \cdot e^{2\pi j\xi t}$$

.

One step further, the $e^{2\pi j\xi t}$ is the eigenfunction for any order derivation operators. The Fourier transform can be understood as projecting a signal from the temporal domain to the spectral domain. Accordingly, the graph transform is defined as

$$\hat{f}(\lambda_l) =< f, u_l >= \sum_i f(i)u_{li}^*$$

.

The connections between the graph transform and Fourier transform can be explained in this way: for Fourier transform, it has a space that is constructed on bases $e^{2\pi j\xi t}$. Each signal $f(t)$ can be represented by these bases, the coefficient of which is exactly $\hat{f}(\xi)$. The bases can be seen as the eigenvectors for the Fourier transform (this is an analogy only since the Fourier transform is operated on a continuous signal). For graph transform, each graph signal f is transformed into the spectral domain, which has bases $\{u_1, u_2, \cdots u_l\}$. The eigenvectors u_l corresponding to low frequency λ_l vary slowly across the graph and vice versa.

Spectral graph convolution

Given the input node features $X \in \mathbb{R}^{N\times 1}$ defined on a graph G, we want to learn a convolutional kernel $H \in \mathbb{R}^{N\times 1}$, the process of the spectral-based GNN is formulated in the following steps:

1. Eigen-decomposition of Laplacian matrix L: $L = U\Lambda U^\top$. Each column of the U is the eigenvector of the L. Λ is a diagonal matrix composed of eigenvalues. The Laplacian matrix of a graph G is decomposed into a set of orthogonal bases. The input node features X can be projected to the space formed with the bases.

2. Project the H and X to the spectral domain using U:

$$\begin{aligned} \hat{H} &= U^\top H, \\ \hat{X} &= U^\top X. \end{aligned} \tag{2.1}$$

3. Spectral multiplication and reprojection:

$$\begin{aligned} X' &= U(\hat{H}\hat{X}), \\ &= U(U^\top H \cdot U^\top X). \end{aligned} \tag{2.2}$$

Where X' denotes the output feature of one GNN layer, $\cdot$ denotes pointwise multiplication. To get deep GNN, multiple layers need to be stacked and rerun the above steps. For the final output node features, there are two kinds of tasks that can be applied. One is the node-level task, such as social media account classification, in which each social media account is represented as one node. The second is the graph-level task, such as text classification, in which each word in the text is seen as one node, and the whole graph can be classified as spam or not.

2.1.2 Methods and applications of spectral-based GNN

For spectral-based GNN, the core work is to design the convolution kernel H. The equation 2.2 can be simplified as $X' = UH_wU^\top X$. There are several methods for designing H_w. [17] set H_w as a diagonal matrix $H_w = diag(w), w \in \Re^N$. This method requires a global convolution of the kernel and the signal and is highly inefficient. [102] proposed a parameterization method to make the kernel spatially localized. [99, 44] proposed a ChebNSet to approximate the kernel H_w up to the K^{th} order. The convolution can then be formulated as

$$\begin{aligned} H_w * X &\simeq \sum_k w_k T_k(\tilde{L})x, \\ \tilde{L} &= \frac{2}{\lambda_{max}} L - I_N, \\ T_k(x) &= 2xT_{k-1}(x) - T_{k-2}(x), \\ T_0(x) &= 1, \\ T_1(x) &= x, \end{aligned} \tag{2.3}$$

where $w_k T_k(\tilde{L})$ denotes the approximation, L is the Laplacian operator, and λ_{max} is the largest eigenvalue. By the summation on k, the ChebNet has a K-localized convolution.

[127] proposed a GCN by simplifying the equation 2.3 with $K = 1$. Besides, for a normalized Laplacian operator, its maximal eigenvalue is 2. So the equation can be simplified as $H_w * X \simeq w_0 X + w_1(L - I_N)X$. Further, constrain the parameter $w = w_0 = -w_1$, we have $H_w * X \simeq w(I_N + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})X$. As stated in [127], renormalization $(I_N + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \rightarrow (\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}})$ can prohibit the gradient vanishing problem. The final form is represented as

$$X' = \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}} X$$

.

Other methods like [147] proposed AGCN which learned adaptive relations among vertices that can be further added to the existing edges. This method makes the graph structure capable of alternating in training. [372] proposed DGCN to capture both local and global consistency using two networks.

2.1.3 Spatial-based GNN

The spectral-based GNN has some limitations; 1. It cannot be applied to the directed graph, 2. the graph structure can not be changed during training, 3. its computation complexity is high. To overcome these drawbacks, spatial-based GNN is put forward. The idea of the spatial-based GNN is to make a direct convolution operation on the graph. This operation can fit the different structures of neighboring vertices and also has local invariance. [60] proposed neural FPs that aggregate neighboring vertices according to their degrees:

$$\begin{aligned} t &= h_v^t + \sum_{u \in N_v} h_u^t, \\ h_u^{t+1} &= \sigma(t W_{N_v}^{t+1}), \end{aligned} \tag{2.4}$$

where $W_{N_v}^{t+1}$ refers to the weight matrix. $|N_v|$ refers to the degrees of vertex v. This method does not suit the graph with large degrees. [98] proposed an inductive framework

for embedding and aggregation for a vertex and its neighbors, named GraphSAGE.

$$\begin{aligned} h_{N_v}^{t+1} &= AGG_{t+1}(\{h_u^t, \forall u \in N_v\}), \\ h_v^{t+1} &= \sigma(W^{t+1} \cdot [h_v^t || h_{N_v}^{t+1}]), \end{aligned} \tag{2.5}$$

where AGG_{t+1} denotes information aggregation, which can be a mean, pooling, or long short-term memory (LSTM) operation. [285] proposed a prevalent network 'graph attention network'. The idea is to aggregate information from the first-order neighbor nodes. The process is formulated in:

$$h_i' = \sigma(\frac{1}{K} \sum_{k,j \sim \mathcal{N}_i} a_{ij}^k W^k h_j), \tag{2.6}$$

a_{ij}^k denotes each entry of the adjacency matrix, W^k denotes the weight parameters, and h_j is the feature from the last layer. [197] proposed a MoNet to unify the framework of the spatial-based GNN. In the Monet, each vertex of a graph can be regarded as the origin of a pseudo-coordinate system. The convolution operation is defined according to two functions f and g.

2.2 Relationship Recognition

In this work, we focus on the inter-object relationships, which describe the cross-object interactions in a scene. Thanks to the boosting performance brought by deep learning, DNN-based methods show promising performance due to their high representation capability [42, 174, 338, 349, 371, 131, 153, 210, 160, 162]. The most related works to ours are the ones that try to detect the relationship in scene graphs as one of the attributes [154, 117, 67, 22, 157]. Here we introduce two approaches for scene graph generation (SGG): two-stage and one-stage.

2.2.1 Two-stage methods

The scene graph generation requires the model to simultaneously detect objects and relationships among them. It is tough since the object detection accuracy would affect the

relationship recognition. Currently, two-stage scene graph methods are prevalent since they unload the object detection to offline models like faster-rcnn [232] which is pre-trained on the large-size dataset. There are many types of two-stage methods. Some popular ones are described below.

Conditional random field based SGG:

The scene graphs were first put forward by [117] and utilized for image retrieval. In the task, a scene graph can represent the interactions between objects in an image, thus providing a better set of retrieval features. [247] proposed to use two parsers, a rule-based and a classifier-based to model the dependency as transition belief in a scene graph. [42] used two potential functions which both model the appearance and predicates of any two objects.

The objective of the SGG is to maximize the function:

$$P(SG|I) = \prod_{o_i \in O} P(o_i, o_i^{bbox}|I) \cdot \prod_{r_{i \to j} \in E} P(r_{i \to j}|I), \tag{2.7}$$

where the I denotes the input image, o_i denotes the i_{th} detected object, and o_i^{bbox} denotes the bounding box. $r_{i \to j}$ denotes the relationship between objects i, j. The first term maximizes the object detection accuracy given the input image, and the second term tries to maximize the relationship recognition.

Knowledge graph based SGG

The knowledge graph is popular in natural language processing to embed the language according to its semantics. It has a similar structure to the scene graph. Combining the pros of the knowledge graph [349], the SGG tries to model the relationship in the form of *(subject-relationship-object)*, also denoted as *(s, p,o)*. The idea is that any s can be translated to o by the relationship p or $!p$ as $s + p \approx o$ or $s + !p \approx o$. Both s, o are represented with parameters W_s, W_o as $s = W_s x_s, o = W_o x_o$. The regularization term, its main purpose is to relieve the noise level in the estimated cost volume computed from the

extracted features.

$$W_s x_s + t_p \approx W_o x_o, \tag{2.8}$$

where t_p denotes translations. The objective is formulated as

$$\mathcal{L} = \sum_{(s,p,o)} -\log[softmax(t_p^\top (W_o x_o) - W_s x_s)]. \tag{2.9}$$

Based on the formulations, [349] proposed to embed the objects' representations to low-dimensional space, and predicates' representations to translation vectors. [111] applied the scene graph to the unseen relationships. In this work, the context information is aggregated to capture rare and unseen relations.

CNN-based SGG

For CNN-based SGG, the relationship is also modeled as the form *(subject-relationship-object)*. Firstly, an object detection model would extract bounding boxes related to objects and then predict relationships given the object features. In this process, some weak connections might be cut off for computation consideration. [42] applied CNN to the scene graph generation. They explored the spatial information and statistical dependency in the *(subject-relationship-object)* tuples. The method can be divided into three steps: 1. Region proposal. In this step, a single image is parsed into different objects with bounding boxes; 2. Filtering. Some unimportant objects or relationships are filtered out according to their spatial information and object categories. 3. Generation. All pairs of objects are processed, as well as their appearances, and statistical dependencies with different predicates. [167] not only used a fixed object detection model but also tried to finetune it in the training. They proposed the SIN network, which considered both visual appearance and contextual information. In the SIN, they adopted the same technique region of interest (ROI) pooling as in the faster-RCNN. So each object is embedded into a feature vector and regarded as the node. After that, both spatial and visual features are aggregated to the edge, which represents the influence between nodes. [350] proposed a relationship proposal network. This model imitates the faster-RCNN to propose relationships, which aims to remove those unimportant relationships and reduce computation complexity. In a scene,

the number of detected objects is N, and the number of $< subject, predicate, object >$ tuples will be $N \times N$. In the work, 9 kinds of relationship proposals are used according to different conditions. So three sets of features of subject, predicate, and object are then connected to get a score. [300] proposed a LinkNet to improve the scene graph generation performance by exploring the inter-dependencies among all detected objects. The LinkNet contains three components: 1. Relation embedding: it proposes all objects and relationships for classification. 2. Global context encoding: it tries to model the influence of the context information. 3. Geometric layout encoding: it encodes the spatial information for better classification.

GNN-based SGG

The scene graph generation has an innate similarity with the GNN. So it is natural to model the scene graph with the GNN. Here an object detection model is first applied. Then all detected objects act as nodes to construct a graph, and the connections among nodes are seen as relationships. By using spectral-based or spatial-based GNN, an image can be modeled as a scene graph.

[112] put forward to using the Language prior, which makes the relationship recognition more consistent with our common sense. It obtains the statistics of predicate distribution and adjusts the probability of the recognized relationships. The model first extracts the objects from the image and then uses an iterative message passing to optimize the graph neural network. Besides, an offline relational module that represents the language priors is used to refine the recognized relationships. [318] proposed a graph RCNN that contains three stages: 1. object detection; 2. relationship proposal; 3. graph context integration. In the beginning, the model used a faster-RCNN for object proposal. Then, a relationship proposal network is utilized to obtain meaningful relationships. Besides, an attentional graph convolutional network is proposed to aggregate relationship representations from neighboring nodes. [8] tried to combine the attention with the transformer. The original transformer was proposed by OPENAI for language processing. Here the author puts the final layer output to the "attention graph" and trains with two cross-entropy losses. The advantage of the method is it can construct the whole graph "holistically" so that all nodes

can be output at once in a parallel way.

2.2.2 One-stage methods

The two-stage methods use an offline object detection model to generate object proposals. However, the proposals are immutable during scene graph generation. So some one-stage methods are proposed to jointly train object detection and relationship recognition. Here the dataset would be an obstacle. Fortunately, the image caption tasks provide a promising solution. [149] simultaneously solve three tasks: object detection, scene graph generation, and image caption in a single model. The model first generates object and caption region proposals, and then feeds them into a dynamic scene graph generation module. Finally, three heads are applied to output object categories, scene graphs, and image captions.

2.2.3 Datasets for scene graph generation

Visual phrase [239] is a dataset for visual phrase recognition. It contains 8 object categories and 17 visual phrases which are formed either by interaction or activities of objects.

Scene graph [117] firstly put forward a scene graph dataset for real-world scenes. It contains 5000 images, and each image contains a human-annotated scene graph.

Visual relationship detection [174] proposed a benchmark for scene graph generation. The work highlighted one important phenomenon, long-tail distributed relationships. The training data contains 4000 images and the test data contains 1000 images. The relationships include action, verbal, spatial, preposition, and comparative ones.

Visual Genome [132] is one of the most prevalent datasets for scene graph generation in current days. It has more relationships and diverse object categories. The dataset is annotated by the crowd work. The advantage is the annotations comply with the majority of our commonsense. The disadvantage is that some parts of the annotations are of poor quality. Some researchers tried to reform the dataset, including VG200 [349], VG150 [310], VG-MSDN [149], VG80K [351], and sVG [42]

Open images [133] offers 9 million images with annotations: image-level labels, bounding boxes for objects, object segmentation masks, visual relationships, and localized narratives. 8.3 objects are contained in each image on average. There are 16 million bounding boxes in total for 600 class objects. It is the largest existing dataset, which pertains to different tasks.

ImageNet-VidVRD[251] provides data for video visual relationship detection. It is constructed on 1000 videos, 35 categories of objects, 3219 relationship triplets, and 132 predicate categories.

VidOR [250] has 10000 videos measuring more than 98.6 hours. It contains 80 objects and 50 predicate categories. There are 7000 videos for training, 835 videos for validation, and 2165 videos for testing. The average length of the video is 35 seconds. The relationships are divided into two types: spatial relationships and action relationships.

2.2.4 Applications of scene graph

Image retrieval. This is an old but important task in computer vision. The task requires the user to give the query, like an image or text description, and the model returns the retrieved images. The scene graph has a good representation of inter-object interactions in a scene, which makes it capable of reasoning about images. Besides, the scene graph is able to ground the object in an image region. For image-based queries, it tries to find the most similar samples from both text and image domains. The similarity is quantified as the score. Both text and image can be constructed with scene graphs. The similarity can be calculated on node-level and edge-level.

Image generation. This task refers to generating complex realistic scenes from the objects and the desired layouts. There are many challenges: 1. how to process the graph-structure data, 2. how to guarantee the uniformity between the generated images and their corresponding graphs, and 3. how to guarantee the fidelity or authenticity of the scene.

Image/Video captioning. The captioning based on the scene graph is different from the traditional one. It has two steps, concept recognition and sentence construction. For language-and-vision tasks, grounding the objects in an image with the text in a phrase is

important. Method [141] tried to use two generators to generate image and text features for visual grounding. Scene graph-based autoencoder is also applied for image captioning [320].

Visual question answering. Scene graphs contain abundant information about the objects such as appearance, shape, attributes, and relationships. So it provides explicit cues for question answering. [348] motivated by traditional methods using knowledge graph, adopts the scene graph as reasoning cues. [87] proposed to use an attention mechanism for visual question answering. The methods try to combine the natural language explanation to answer a question.

Image understanding and referring To understand an image, it needs to be parsed into the form of (objects, actions, scenes). Some high-level understandings need to be extracted thereby. So a scene graph can provide priors for reasoning. [5] proposed a scene description graph to construct a perception system. In this work, the common-sense knowledge is constructed with a Bayesian network on objects' occurrence frequency.

2.2.5 Methods statistics

Methods	VTransE	Vip-cnn	Graph-rcnn	RelDN	MotifNet	VidVRD	VCTREE	self-supervision
Performance (R@50)	14.07	17.32	11.40	28.30	27.20	5.54	27.90	28.28
Dataset	VRD	VRD	VG150	VG150	VG150	ImageNet-VidVRD	VG150	VG150
Year	2017	2017	2018	2019	2018	2017	2019	2020

Table 2.1: Statistics of some state-of-the-art methods.

2.3 Depth Estimation

Early works of depth estimation [244, 69, 68] focus on utilizing geometry-based information, such as the points correspondence among different views. With the development of deep learning techniques, the CNN based depth methods [62, 61, 135, 19, 72, 16] show superior performance on depth estimation tasks. Some of them try to improve the performance by

avoiding using the ground-truth information from the real world [234, 366, 9, 34, 363, 80, 89, 369, 287, 226, 73, 71, 74, 161], which reduce the demand for expensive annotations. Other works aim to increase the methods' generalizability and work in the wild [376, 29, 302, 30, 303, 155, 190] thus saving the heavy training cost. The most related works to ours are the multi-task ones[106, 158, 120, 134, 61, 291, 145, 234, 125, 214, 309, 361, 362, 27, 175, 159], where information from other tasks are used in depth estimation. However, none of them utilizes the spatial priors in inter-object relationships. In this section, we introduce depth estimation from two aspects: stereo-based depth estimation and monocular depth estimation.

2.3.1 Stereo-based depth estimation

For stereo-based depth estimation, the basic method is to estimate by stereo matching. The method would take $n = 2$ images as input and output a disparity map D. The objective function is

$$E(D) = \sum_x C(x, d_x) + \sum_x \sum_{y \in N_x} E_s(d_x, d_y), \tag{2.10}$$

where x, y denotes pixels and N_x denotes the neighboring pixels of x. The first term measures the cost of matching pixel x in the left image to pixel y in the right image. In the case that the depth level is discretized into n_d levels, the $C(x, d_x)$ represents a 3D cost volume with size $H \times W \times n_d$. The second term provides regularization for the estimation. The general framework for stereo matching contains four steps, 1. feature extracting, 2. feature matching, 3. disparity computation, and 4. post-processing. [345, 100, 347] applied CNN to take two patches both from the left and right images to extract features. Then a matching module would be applied to the two patches to evaluate the similarity. One commonly used is smoothness regularization:

$$E_s(d_x, d_y) = \alpha_1 \delta(d_{xy} = 1) + \alpha_2 \delta(d_{xy} > 1), \tag{2.11}$$

where $d_{xy} = d_x - d_y$ and δ is Kronecker delta function.

Another method uses an end-to-end framework. The early works like FlowNet-Simple [52] and DIspNetS [186] applied encoder-decoder to stack left and right images to a 6D

cost volume. The advantage of this type is it doesn't explicitly require stereo-matching steps but needs a lot of data.

2.3.2 Monocular depth estimation

The MDE is that humans can perceive depth by the motion parallax and object layouts in a single image. So a single image provides abundant depth cues. Some deep learning-based works try to solve the problem by exploring priors.

Supervised method. For the supervised training, the MDE problem can be seen as a regressive problem [62]. So a deep neural network could be applied to it. The objective is:

$$L_2(d, d^*) = \frac{1}{N}\sum_{i}^{N} ||d - d^*||_2^2, \tag{2.12}$$

where d denontes the predicted depth, d^* denote the ground-truth depth. The framework of the [62] is the most well-known for MDE. It is composed of two components. The whole model is trained in an end-to-end manner. The objective is formulated as

$$L(d, d^*) = \frac{1}{N}\sum_{i}^{N} y_i^2 - \frac{\lambda}{N^2}(\sum_{i}^{N} y_i)^2$$

where $y_i^2 = \log(d) - \log(d^*)$. In this way, the objective is to estimate the log depth of the image. [61] proposed a network that is able to estimate the surface normal, depth map, and semantic label from a single image input. The additional objective is formulated as

$$L_S = \frac{1}{N}\sum_{i}^{N}[(\nabla_x D_i)^2 + (\nabla_y D_i)^2], \tag{2.13}$$

where $D_i = \log(d_i) - \log(d_i^*)$, ∇ denotes the differential operator. This term constrains the output to have similar gradients as the ground truth. [186] extended the model and applied it to the scene flow estimation. [252] proposed to use a fully CNN-based framework for depth estimation. [135] proposed to use Resnet for depth prediction, so that the model was much deeper than others and had better performance. [181] put more attention on

the application of depth in obstacle detection. The proposed model used both monocular images and optical flow.

Self-supervised method. Unlike supervised learning, where the model uses the ground truth as the supervision, the self-supervised methods use the geometric constraints between frames as supervision. These methods are trained with monocular scene sequences. The geometric constraints are formulated as

$$p_{n-1} \sim K T_{n \to n-1} D_n(p_n) K^{-1} p_n, \tag{2.14}$$

where the p_n denotes the pixel in an image I_n, K is the camera intrinsic matrix. $D_n(p_n)$ denotes the depth map at the pixel p_n. $T_{n \to n-1}$ denotes the transformation between the images I_n, I_{n-1}. [272] use a view synthesis as a metric, and the objective is represented as

$$L_{vs} = \frac{1}{N} \sum_{p}^{N} |I_n(p) - \hat{I}_n(p)|, \tag{2.15}$$

where p denotes pixels in the reference frame and the $\hat{I}_p$ denotes the reconstructed frame. Besides, the structure similarity loss is formulated as the objective

$$L_{vs} = \alpha \frac{1 - SSIM(I_n - \hat{I}_n)}{2} + (1 - \alpha)|I_n - \hat{I}_n|. \tag{2.16}$$

Besides, [90] also proved it is more effective to use the minimal value rather than the mean to construct the objective.

Conditional random fields method. Except for the methods using supervised or unsupervised ways, some researchers focus on conditional random fields (CRF) for depth estimation. In this way, the output depth map by the network would be further refined by the conditional random fields. The depth map has the property of continuous value, so the depth of each pixel can be refined by its neighbors. [145] use a deep CNN to estimate the depth map at the super-pixel level. Then the depth map is refined by

$$E(d) = \sum_{i \in S} \phi_i(d_i) + \sum_{(i,j)} \in \varepsilon_s \phi_{i,j}(d_i, d_j) + \sum_{c \in P} \phi_c(d_c), \tag{2.17}$$

where S denotes the set of super-pixels, ε_s denotes the pair of super-pixels. The first term measures the distance between the original and the ground truth. The second term

constrains the smoothness between super-pixel pairs. The third term describes the local relevant structures.

Graph-related methods. Some works have investigated extracting information from graph structure to enhance the accuracy of the estimated depth. We present a detailed review of them to compare with our proposed method.

GCNDepth [185] is a self-supervised monocular depth estimation method based on GCN. It uses a CNN-GCN autoencoder to learn depth features from unlabeled videos and a pose network to estimate camera motion. It can generate accurate and detailed depth maps that preserve object boundaries and discontinuities. The advantages of GCNDepth can be summarized as: 1. It uses a single-stage CNN-GCN autoencoder instead of two consequent networks, which reduces the complexity and computational cost of the model; 2. It uses GCN as a decoder network to improve the detection of sharp boundaries, reduce the background noise, and compute precise depth maps with full object details compared to other self-supervised models; 3. It achieves better quantitative and qualitative performance on depth estimation than existing methods on several benchmark datasets. The workflow of GCNDepth is summarized as: (1) It takes a single RGB image as input and passes it through a CNN encoder network to extract a low-resolution feature map. (2) the feature map is upsampled using bilinear interpolation and reshaped into a matrix where each row represents a node in the graph. (3) it constructs an adjacency matrix that represents the connections between nodes based on their spatial proximity and color similarity. (4) An adaptive edge is applied to assign different weights to different edges based on their importance for depth estimation. (5) The node matrix and the adjacency matrix are passed through a GCN decoder network that consists of several graph convolutional layers and skip connections from the encoder network. (6) Adepth map that preserves the global context and local details of the scene is output. The GCNDepth distinguishes itself from our proposed method in the following aspects: (1) We explicitly introduce the relationship understanding among objects into the depth estimation, which is a high-level understanding of the scene structure. However, the GCNDepth only models the connections by weighting the edges among nodes in a graph. (2) Our method is able to quantify the contribution of each relationship while the GCNDepth only quantifies the importance of each row of the image which is not related to the scene structure understanding. (3) We also claim

to explore spatial information entailed in relationship understanding, which is out of the GCNDepth consideration.

Neural Window Fully-connected CRFs for Monocular Depth Estimation [341] proposes a novel method for monocular depth estimation based on neural window fully-connected CRFs, which consists of two stages: a depth regression stage and a CRF optimization stage. The depth regression stage uses a ResNet-50 encoder and a decoder with skip connections to generate an initial depth map. The CRF optimization stage uses a neural window function to define the pairwise potentials between pixels and optimize them with mean-field inference. The neural window function is learned by a small network that takes the image features and coordinates as inputs and outputs a scalar value for each pixel pair. The advantages are summarized as: 1. This method leverages the potential of fully connected CRFs to model the long-range dependencies between pixels and refine the depth map. 2. This method splits the input into windows and performs CRF optimization within each window, which reduces the computation complexity and makes the method feasible. 3. This method exploits the multi-head attention mechanism to compute a multi-head neural window function that captures the relationships between nodes in the graph. 4. The paper shows that the proposed method can handle challenging scenes with complex structures, occlusions, and textureless regions. The method distinguishes itself from our proposed method in the following aspects: 1. The method utilizes CRFs to model the dependencies among pixels. In our proposed method, we try to impose the influence of relationships to pixels by relationship spatialization. 2. The method uses multi-head attention to model the relationships between nodes, While our proposed method explicitly applies the relationships among objects which achieves a high-level understanding of the scene structure. 3. We dig out spatial information contained in explicit visual relationships which is out of consideration of this method.

Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos [35]. The paper proposes a novel method for 3D multi-person pose estimation in monocular videos that combines graph and temporal convolutional networks. It uses GCN to model the spatial relations among human joints and body parts and to estimate the initial 3D poses from 2D poses. The temporal convolutional networks (TCN) are used to model the temporal and human-dynamics constraints across frames and to refine

the 3D poses by estimating the joint velocities. Its advantages are summarized as: 1. It leverages both spatial and temporal information to improve the accuracy and robustness of 3D pose estimation. 2. It introduces a novel velocity-TCN that estimates the joint velocities and enforces the consistency of 3D pose estimation in consecutive frames. 3. The method outperforms existing methods on several benchmark datasets, especially on challenging scenarios such as occlusion, self-contact, and fast motion. Its workflow includes three steps: 1. In the first step, the method uses a state-of-the-art 2D pose estimator to detect and track multiple people in a monocular video and obtain their 2D joint locations. 2. In the second step, the method uses a GCN to estimate the person-centric 3D poses for each person from their 2D poses. The GCN captures the spatial dependencies between joints and learns a mapping function from 2D to 3D space. 3. In the third step, the method uses two TCNs to refine the 3D pose estimation further. The first TCN, called joint TCN, estimates person-centric 3D poses across frames and enforces temporal smoothness and human-dynamics constraints. The second TCN, called velocity-TCN, estimates the joint velocities and enforces consistency of 3D pose estimation in consecutive frames. This paper mainly contributes to estimating the 3d poses for multiple people, which leverages spatial information out of people interactions. Our proposed method distinguishes from it since we explicitly add the interactions, namely the visual relationships to the depth estimation.

2.3.3 Datasets for Stereo-based depth estimation

There are a good number of datasets for depth estimation for training, evaluation, and testing purposes. Some important ones are given in the table 2.2.

2.4 Pedestrian trajectory prediction

In the automotive industry, pedestrian trajectory prediction (PTP), or pedestrian intention estimation (PIE) have been an important task, which helps auto vehicles monitor the road situation and make corresponding adjustments. Due to the development of the deep learning community, deep models have become more and more prevalent in the literature.

	Year	Type	Task	Resolution	Scenes	GT frames	Depth type
Make3D [241]	2009	Real scene	Monocular depth	2272 × 1704	534	534	Dense
Kitti2012 [82]	2012	Real scene	Stereo	1240 × 376	389	-	Sparse
MPI Sintel [18]	2012	Synthetic	Optical flow	1024 ×436	35 videos	-	Dense
NYU2 [199]	2012	Real indoor	Monocular depth	640 ×480	464 videos	1449	Kinect depth
RGBD-SLAM [266]	2013	Real scene	SLAM	640 ×480	19 videos	-	Dense
SUN3D [305]	2013	Real rooms	Monocular depth	640 ×480	415 videos	-	Dense
Middleburry [243]	2014	Indoor	Stereo	2948 × 1988	30	30	Dense
Kitti2015 [189]	2015	Real	Stereo	1242 *times* 375	400	-	Sparse
CityScapes [40]	2016	Street scenes	Dense depth	2048 × 1024	5K	-	-
DTU [1]	2016	Real small objects	MVS	1200 *times* 1600	80	-	structured light
ETH3D [246]	2017	Real in/outdoor	Stereo	940 ×490	47	47	Dense
SUNCG [261]	2017	Synthetic	Scene completion	-	45K	-	Dense
MVS-Synth [108]	2018	Synthetic	MVS	1920 ×1080	120	-	Dense
MegaDepth [151]	2018	Real	Monocular	1600 × 1600	130K	100K	Dense
Jeon and Lee [114]	2018	Real	Depth enhancement	-	4K	4K	Dense
Omni-things [299]	2019	Synthetic	MVS	800 × 768	10240	-	Dense
HR-VS [316]	2019	Synthetic	Stereo	2056 × 2464	780	780	Dense
Driving Stereo [317]	2019	Driving	Stereo	1762 ×800	182188	182188	Sparse
AppIloScape [110]	2019	Auto driving	Stereo	3130 ×960	5165	5165	LIDAR
A2D2 [86]	2020	Auto driving	Stereo	2.3M pixels	41277	-	LIDAR

Table 2.2: Datasets for depth estimation.

The PTP or PIE task refers to predicting the future trajectory, which is represented as the coordinates of pedestrians according to the present trajectory. It involves the processing of temporal images or videos. There are three types of techniques: recurrent neural network (RNN), convolutional neural network-based (CNN-based), and generative adversarial network-based methods (GAN-based).

2.4.1 RNN-based methods

Since the pedestrian trajectory is a temporal sequence of the coordinates, it is tightly related to long-term information processing, where the PTP stands out. [6, 270, 70, 208, 43, 307] proposed using LSTM to model the temporal relations in trajectory. [6] claims each human's trajectory would be influenced by the others', and the hidden states in LSTM can be shared among the pedestrians with neighboring humans. [270] tries to learn human activity patterns thus predicting future trajectory. The problems of the LSTM are that it cannot simultaneously model the spatial interactions of pedestrians and suffers from

gradient vanishing. [43, 307] tries to model the spatial-temporal information.

2.4.2 CNN-based methods

In trajectory prediction, spatial information plays an important role, so CNN is another good choice for modeling. [229, 105, 364, 327, 51, 184, 196, 292] adopt different CNN architectures to solve the PTP. [229] proposed to use CNN to process image signals for motion prediction and a Markov decision process to infer the pedestrian's goal. [105] combined CNN and Bayesian estimation to predict long-term motion and current environment dynamics. [364] used multi-agent tensor fusion and an encoder-decoder architecture to predict multiple pedestrians' trajectories. [327] tended to use crowded scene video data for prediction. [51] tried to predict average occupancy maps of all pedestrians and the results were better than other baselines. [184] applied a memory-augmented neural network which dug out the connection between the past and future trajectories based on memory. [196] used graph representations to improve prediction. [292] modeled the interactions between the pedestrians and the background.

2.4.3 GAN-based methods

Adversarial learning is also a powerful way to handle the PTP task. In this way, the distribution of the coordinates is learned and generated. [65, 96, 129, 7, 130] adopted generative models to predict the trajectory. [65] used a generator to output a probability map for each pixel, which is further processed for trajectory prediction. [96] thought all pedestrians influence each other in a uniform way, including scene context. So they applied a socially aware generative model to predict the trajectory. [129] modeled multimodal behavior of pedestrians in a graph attention network and predicted path with a generative model. [7] applied Info-Gan [31] for the task.

2.4.4 Methods comparison

State-of-the-art methods for pedestrian trajectory prediction are shown in Table.2.3, where the ADE and FDE denote average distance errors and final distance errors.

Methods	Year	Dataset	Link	ADE	FDE
NSP-SFM [344]	2022	ETH-UCY SDD	→	0.17	0.24
Y-Net [182]	2020	ETH-UCY SDD	→	0.18	0.27
NPSN [12]	2022	ETH-UCY	→	-	-
SGCN [253]	2021	ETH-UCY	→	0.19	0.35
PEC-Net [183]	2020	ETH-UCY SDD	→	0.29	0.48
Social-gan [96]	2018	ETH-UCY	→	0.343	0.686
Social-STGCNN [196]	2020	ETH-UCY	→	0.44	0.75

Table 2.3: State-of-the-art methods for the pedestrian trajectory prediction.

2.4.5 Datasets for the PTP

ETH [208] and UCY [142] are the most prevalent methods for the PTP. They are captured with pedestrians walking along the footpath and annotated with location coordinates. The UCY dataset also has annotations of gaze directions to capture view angles. There are totally five scenes in two datasets including ETH, HOTEL, UNIV, ZARA1, ZARA2. There are more than 1600 pedestrian trajectories, which are sampled every 0.4 s. The general data split has a leave-one-out strategy for cross-validation. Another dataset is Standford Drone [235], of which all images are captured by drones. The annotations include class labels of objects and bounding boxes. The location is annotated as the coordinate every 0.4 s. There are more than 19K walking agents recorded and diverse interactions between pedestrians, cyclists, skateboarders, and buses.

2.5 3D Modeling

3D modeling has been an important part of the computer graphics area. This part will review it in two aspects: 1, 3D neural rendering; and 2, generative 3D modeling.

2.5.1 3D neural rendering

3D neural rendering involves representing a natural scene, such as a set of unordered or multi-view images, using classic computer graphics techniques. It separates the camera-capturing process from scene representation by using established image formation models, such as rasterization, point splatting, or volumetric integration. In this section, we will first introduce the formulation and basic concepts, followed by a summary of the most effective current methods.

Scene representation. The 3D scenes are always represented with surfaces or volumetric, where the surface representation always stores the properties of an object's surface, such as coordinates, and curvature, and the volumetric representation stores properties such as densities, and opacities. The surface can be formulated in either an explicit or implicit way. For explicit formulation, it is defined as

$$S_{explicit} = \left\{ \begin{pmatrix} x \\ y \\ f_{explicit}(x,y) \end{pmatrix} \;\middle|\; \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \right\}, \tag{2.18}$$

where $f_{explicit}$ denotes an explicit surface function. The explicit surface can also be represented as a parametric function $f_{parametric} \in \mathbb{R}^3$ as

$$S^*_{explicit} = \left\{ f_{parametric}(u,v) \;\middle|\; \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^2 \right\}, \tag{2.19}$$

The implicit surface representation is formulated with the zero-level set of surface function $f_{implicit} \in \mathbb{R}$ as

$$S_{implicit} = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3 \;\middle|\; f_{implicit}(x,y,z) = 0 \right\}, \tag{2.20}$$

A volume representation is defined as

$$V = \left\{ f_{vol}(x, y, z) \;\middle|\; \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3 \right\}, \tag{2.21}$$

In practice, the surface can be represented as point clouds, meshes, or implicit surfaces such as signed-distance-function. The volumes can be represented as voxel grids or neural volumetric representations.

Rendering techniques. A rendered image of a certain view of a scene can be calculated given the camera pose, most methods use a pinhole camera. Generally used techniques are "ray casting" and "rasterization". The "ray casting" relies on the pinhole model which describes how a 3D coordinate is projected into a 2D image plane given the known camera parameters. The parameters are organized as intrinsic matrix $\mathbf{K}$, including focal lengths normalized by pixel size $\mathbf{f} = [a_x, a_y]$, axis skew γ, and center point $\mathbf{c} = [c_x, c_y]$. With the parameters, a homogeneous coordinate $\mathbf{p} = [x, y, z, 1]$ is projected into 2D plane as $\mathbf{q} = \mathbf{K} \cdot \mathbf{p}$. To generalize to arbitrary camera poses, an extrinsic matrix $\mathbf{R}$ is used. A corresponding homogeneous matrix is defined as

$$\mathbf{E} = \begin{bmatrix} \mathbf{R}_{3\times3}\mathbf{t}_{x\times1} \\ \mathbf{0}_{1\times3}1 \end{bmatrix}, \tag{2.22}$$

where $\mathbf{R}$ and $\mathbf{t}$ denote rotation matrix and translation vector respectively. So we have $\mathbf{R} \cdot \mathbf{p}_w + \mathbf{t} = \mathbf{P}_c$, where $\mathbf{P}_w$ and $\mathbf{P}_c$ denote a point in world coordinates and camera coordinates. The projection form 3D world coordinates to 2D image coordinates is defined as

$$\mathbf{q}_p = \mathbf{K} \cdot \begin{bmatrix} \mathbf{Rt} \\ \mathbf{0}_{1\times3}1 \end{bmatrix} \cdot \mathbf{p}_w, \tag{2.23}$$

where $\mathbf{q}_p$ denotes a point in 2D image coordinates.

"Rasterization" is mostly used with meshes, which are described by a set of vertices $\mathbf{v}$ and faces $\mathbf{f}$. The surfaces are obtained by connecting vertices. In practice, the same extrinsic matrix $\mathbf{E}$ is used to transform each point from the world to the camera coordinate system. The points outside the view frustum will be culled for acceleration. The face

information can be used to get the depth maps. This method is faster than "ray casting" since it only scales with the number of visible vertices.

2.5.2 Applications

Novel view synthesis of static scenes is the task of rendering a given scene from new camera positions. Deepstereo [68] proposed the first end-to-end training pipeline for novel view synthesis. In this method, input images are presented with a plane sweep volume, where each pixel is reprojected to a depth plane, and the neural network evaluates the accuracy of the reprojection. However, the Deepstereo is slow in rendering because it needs to estimate the depth for each input image. Stereo Magnification [370] improves the inference speed by directly representing a plane sweep volume into a 3D voxel grid. However, this method only renders novel views in a small view variance. MPI [263] further improves it by adding more input images. DeepVoxels [258] learn a renderer with a 3D voxel grid of features. Similarly, Neural Volumes [171] adopts a 3D CNN to learn a volumetric representation of a scene. To overcome the storage limitations brought by these methods, SRNs [259], DVR [203] apply a MLP to represent the scene. NeRF [193] made a breakthrough in obtaining photo-realistic rendering for MLP-based represented scenes. It can generate high-quality renderings with a few amount of parameters, which is more efficient than voxel grid representation. However, this method suffers from the problem of low rendering speed. To accelerate the rendering speed, Neural Sparse Voxel Fields [164] applied an octree structure that allows aggressive empty space skipping. KiloNeRF [230] designs a structure combining a dense 3D grid of MLPs, where each MLP is much smaller than a standard NeRF model.

Generalization over scene classes. Since the NeRF-based model overfits a single scene with an MLP model, it is hard to generalize to different classes. PixelNeRF [333] leverages an image encoder to extract pixel-wise features with a volume rendering framework. The novel views are rendered by aggregating these extracted semantic features. GRF [281] improves the rendering quality by designing an additional attention module. Stereo Radiance Fields [36] similarly extract semantic features but learn correspondence matching across multiple images. LOLNeRF [228] learns to generalize with portrait im-

ages, which are conditioned on instance-specific latent vectors. GeoNeRF [116] constructs a set of cascaded cost volumes and employs transformers to infer geometry appearance.

Novel view synthesis of non-static scenes strive to learn time-varying representations to render novel views for dynamic scenes. Since the original NeRF only learns field representations for static 3D spatial points, [150, 304, 77] extend it by learning the deformations of the pixels with warped rays. They also applied motion flow information to improve the spatial and temporal consistency of the rendered dynamic scenes. Moreover, these methods also investigate representing the foreground and background contents separately. D-NeRF [212] designs an unregularized ray-blending MLP to learn the deformations of the ray. HyperNeRF [206] applies a canonical hyperspace to handle scenes with topological changes. Non-rigid NeRF [280] learns a per-scene canonical volume to model the time-varying appearance.

2.5.3 Generative 3D modeling

We introduce two types of generative methods for 3D modeling: (1) optimization-based generation and (2) generative novel view synthesis.

Optimization-based generation refers to the generation involved in runtime optimization. These methods integrate textual or visual prompts with 3D generation to achieve better controllability. TANGO [33] employs CLIP [221] to supervise differentiable physical-based rendering to align text prompts with the generated images. DreamFusion [211] combines the implicit neural field representations with the score distillation sampling loss to generate high-quality 3D content. Magic3D [156] employs a coarse-to-fine pipeline to extract the underlying geometry as a 3D mesh. ProlifcDreamer [295] proposed that the variational score distillation improves the text-to-3D generation. It designs a gradient-based optimization pipeline inspired by the Wasserstein gradient flow. MVDream [254] adopts a two-stage pipeline and fine-tunes a multi-view diffusion model, which overcomes the multi-face and content-drift problems. Other works rely on visual prompts, i.e., images, to improve generated content quality. NeuralLift360 [311] used an offline depth estimation model to impose geometry regularization and further CLIP diffusion priors to enhance appearance. RealFusion [188] and Nerdi [47] extract textual embeddings as conditions to

optimize volumetric representations. Magic123 [215], inspired by Magic3D, integrates 3D priors from a pre-trained Zero-1-to-3 [166] model in a two-stage optimization way. Subsequently, Make-it-3D [277] further enhances the second stage to generate higher-quality point-clouds data. With the popularity of 3D Gaussian Splatting [123], many works have started investigating its generation capacity. Dream-gaussian [276] proposes a two-stage generative framework for text- and image-driven 3D generation. We summarize part of the methods of 3D generation in Table 2.4, evaluated by the quality of reconstructed surfaces and synthesized novel views.

Methods	Chamfer Distance ↓	Volume IOU↑	CLIP Similarity↑	PSNR↑	LPIPS↓
RealFusion [188]	0.0819	0.2741	0.735	20.216	0.197
Magic123 [215]	0.0516	0.4528	0.747	25.637	0.062
Make-it-3D [277]	0.0732	0.2937	0.839	20.010	0.119
one-2-3-45 [165]	0.0629	0.4086	0.788	23.159	0.096
Zero-1-to-3 [166]	0.0339	0.5025	0.759	25.386	0.068
SyncDreamer [168]	0.0261	0.5421	0.837	25.896	0.059

Table 2.4: Methods and Evaluation of 3D generative methods. ↑ denotes the higher the better, while ↓ denotes the lower the better.

Generative Novel View Synthesis involves generating novel views by implicit 3D-aware model, instead of explicitly using 3D representations to enhance 3D consistency. 3DiM [297] pioneer introducing an image-to-image novel view synthesis method based on 2D difussion, which uses view camera poss as condition. Tsenget al. [284] further improves the quality of synthesized novel views with epipolar attention layers. Chan et al. [21], and Yoo et al. [331] apply pixel-aligned features extracted from 2D images as conditions to improve 3D-awareness. In summary, the generative novel view synthesis is recognized as an extension of traditional image synthesis methods, which don't explicitly model 3D structures. It needs more refinement for 3D generation.

2.5.4 Datasets for Generative 3D methods

With the rapid development of digital content generation methods, various types of datasets have emerged. In this section, we will introduce the datasets and their properties. The datasets can be divided into three types: 1, 3D data; 2, multi-view data; and 3, single-view data. The 3D database includes data collected from RGB-D sensors, Polygon models collected from the internet, and published 3D models by certain institutions. The multi-view database mainly includes multi-view images generated by manual 3D Modeling. Single-view images generally contain specific domain data, such as the face, indoor scene, and human body. We conclude the datasets as shown in Table 2.5

Dataset	Type	Year	Samples	Category
ShapeNet [23]	3D Data	2015	51K	Objects
Thingi10K [368]	3D Data	2016	10K	Objects
3D-Future [71]	3D Data	2020	10K	Furniture
GSO [53]	3D Data	2022	1K	Household items
Objaverse [46]	3D Data	2022	800K	Objects
OmniObject3D [301]	3D Data	2023	6K	Objects
Objaverse-XL [45]	3D Data	2023	10.2M	Objects
ScanNet [41]	Multi-view Images	2017	2.5M images	Indoor scenes
CO3D [231]	Multi-view Images	2021	1.5M images	Objects
MVImgNet [340]	Multi-view Images	2023	6.5M images	Objects
DeepFashion [170]	Single-view Images	2016	800K	Clothes
FFHQ [119]	Single-view Images	2018	70K	Animal faces
AFHQ [39]	Single-view Images	2019	15K	Human faces
SHHQ [75]	Single-view Images	2022	40K	Human bodies

Table 2.5: Datasets for 3D data generation.

2.6 Implicit Neural Representation

In this section, we progressively introduce the implicit neural representation starting with the background, exploring its generalization to large-scale data, and the integration into generative models. Additionally, we clarify the connections between these methods and our proposed model.

2.6.1 Implicit Neural Representation

The commonly used data such as audio, image, video, or 3d mesh is represented in signal space. These signals can also be formulated as a mapping from coordinate to signal: $f : C \to Y$, where C denotes coordinates, and Y denotes the signal. f is referred as a *field representation* [306]. In [84, 85, 264] the f is defined with parameterized MLP $f_{\boldsymbol{\theta}}$ which is called *INR*. Parameterizing the signal into an MLP enables the unified representation of various data modalities. For example, audio data can be represented as $f_{\boldsymbol{\theta}} : \mathbb{R} \to \mathbb{R}$ [267, 377, 137, 176]. An RGB image can be formulated as $f_{\boldsymbol{\theta}} : \mathbb{R}^2 \to \mathbb{R}^3$ [257, 64, 240, 312, 32] where the image's resolution can be arbitrarily changed simply by down/up-sampling the coordinates [200, 353]. Besides, the compactness of the INR also achieves remarkable compression performances [265, 56, 95, 360, 25, 315, 152]. Moreover, 3D data such as point cloud, neural radiance fields, and neural distance fields can be represented as $f_{\boldsymbol{\theta}} : \mathbb{R}^3 \to \mathbb{R}^3$ [365, 237], $f_{\boldsymbol{\theta}} : \mathbb{R}^5 \to \mathbb{R}^4$ [193, 13, 14], and $f_{\boldsymbol{\theta}} : \mathbb{R}^3 \to \mathbb{R}$ [37, 205, 279]. Despite the INR achieving good performance on different data modalities, it is hard to generalize large-scale data, because the training of INR overfits each data instance with a separate MLP. Many works strive to improve INR's generalization, which we introduce next.

2.6.2 Generalized Implicit Neural Representation

Research on generalizing the INR to new content can be divided into two types. 1. Learn general input features rather than fixed coordinates; 2. Modulate the parameterized network with hyper-latent. The generalizability of the first type method counts on the general input features. Daniele [94] *et. al.* embed the input coordinates from Euclidean space to

graph-structured non-Euclidean space, which enables representing different data on the same irregular domain with the same INR network. [238] learns the INR on the spherical harmonics basis functions, achieving comparable performances as positional encoding [257]. pixel-nerf and similar works [333, 107, 138] employ an encoder to learn the general features of input signals. However, these methods require the accessibility of signal fractions. In our proposed method, each signal can be generated from a noise sample, not requiring any signal information. The second type of method modulates the weight parameters of INR via latent vectors. Kim *et. al.* [124] construct an instance composer to generate a latent vector for each data instance. Functa-like methods [57, 58, 97, 187, 209] adopt a hyper-network structure and modulate the parameters with scale or shift. The training of the modulations counts on meta-learning methods such as MAML [66] or CAVIA [374]. However, these methods don't consider the hierarchical structure of latent vectors, which prohibits the controllability of data generation.

2.6.3 Generative Implicit Neural Representation

The generative INR distinguishes from the generalized INR on that the former generate new content from noise samples which always involves with generative models such as GAN [92] and latent diffusion model [104, 260], the later only learns the general model which requires fine-tuning to generate new content. The generative INR has two branches of studies: 1. count the generation on the input features; 2. learn to generate new INR parameters in a two-stage manner. For the first type, GIRAFFE [202], GRAF [248] learn a generative model from noise to shape and appearance codes. Combining with sampled coordinates, the codes are sent to a fixed INR to generate new content. Eric *et. al.* learn a generative INR via the generative modeling of input features on a conditional diffusion model. Dream-series methods [254, 224, 168, 328] learn an end-to-end generative INR model from input coordinates to output signals. However, these methods restrictively work on the fields representation which loses the universal representation property of INR. For the second type methods, Dupont *et. al.* [59, 57] model the distribution of the latent vectors which modulate the INR parameters. By sampling new latent vector, the new data instance can be generated. MNIF [332] enhances the expressivity of INR based on [57] with

a mixture of experts architecture. Hyper-diffusion [63] employs a transformer architecture to construct the generative model to learn the distribution of latent vectors. However, these methods doesn't consider the conditional dependency in the INR parameters. The disribution modelling is thus inaccurate and the generation process is hard to control. In our method, we adopt the similar two-stage pipeline as [57] and propose a hierarchically conditional diffusion model to model the conditional distribution innate in the latent vector, thus achieving better generative performances.

Chapter 3

Perceptual Relationships for 3D Understanding: 2D-image based monocular depth estimation

3.1 Introduction

In extracting 3D understanding from monocular 2D images, human perceptual understanding provides valuable spatial priors for inferring 3D information. For instance, estimating object depth is aided by understanding inter-object relationships, which inform relative positioning. Thus, the ability to extract perceptual cues, such as inter-object relationships, is crucial for enhancing 3D understanding. This chapter focuses on extracting relationship representations to improve the accuracy of MDE.

While understanding inter-object relationships, such as *in front of* and *behind*, provides valuable spatial priors, identifying which relationships are most beneficial for depth estimation and quantifying their contributions to the process remains a challenge. In this chapter, we term the task of answering these two questions as 'Relationship Spatialization' for Depth Estimation. Having a look at Fig. 3.1, it can be found that the woman is closer to us than the screen because it is *in front of* the screen, which provides explicit spatial



Figure 3.1: Given a scene image (a), relationships contribute to depth estimation, as shown left to right in (b-d). Compared with the explicit spatial information from *in front of* (a), the implicit spatial information from *under* (b) also benefits the depth estimation. More interestingly, *wear* (c), which is intuitively non-spatial, in reality, contributes significantly to depth estimation.

prior. The table is farther than the bag, although the spatial prior of *under* is implicit, the bag has with smaller depth in the imaging coordinate because it is closer to the image center compared with the table. More interestingly, although *wear* seems with no spatial priors intuitively, it certainly contributes to the depth estimation, since it constrains that the depth of the shirt should be aligned with the depth of the man. Hence, it can be seen that the inter-object relationships provide a huge amount of spatial priors for depth estimation.

Towards solving the proposed Relationship Spatialization task, we propose a novel framework that combines both visual understanding and relationship understanding for depth estimation. Different from the simple combination of information, the relationship representations are automatically and adaptively weighted, thus are learned to be effectiveness-identified and contribution-quantified. Thus, the relationships are spatialized for depth estimation. Specifically, given a scene image, we first feed it into an image representation extractor module, which can be any deep learning MDE model, to obtain the image representations. Meanwhile, the objects in the image are detected to construct the corresponding scene graph, where nodes denote the objects and edges denote the inter-object relationships. Then, the scene graph is processed by the relationship recognition module to predict the inter-object relationships and extract their corresponding represen-

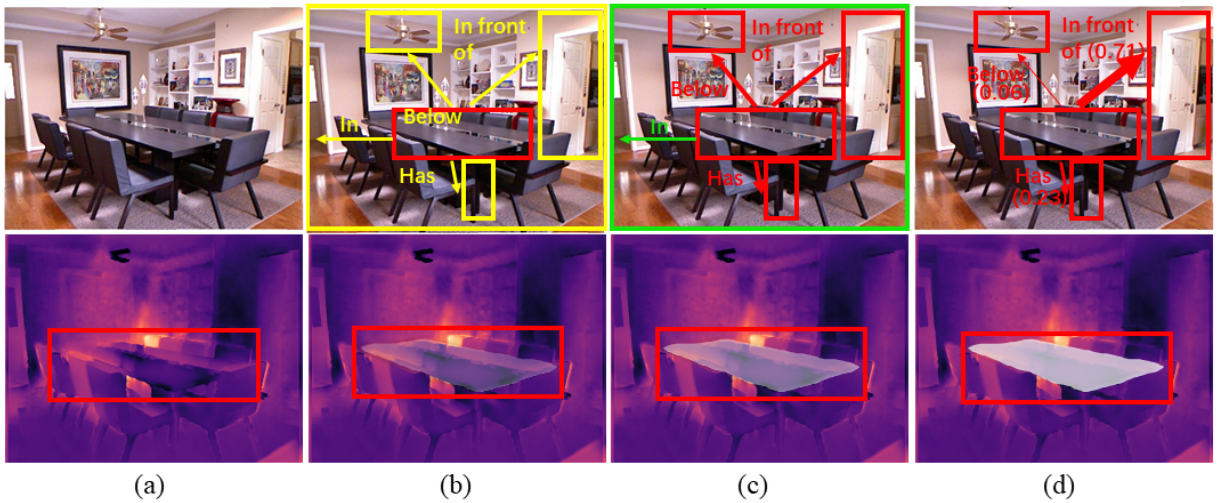


Figure 3.2: Depth error decreases during our relationship spatialization process in the red boxed Table area (a-d). Only the visual understanding is utilized in (a); all Table-related in-image relationships are added in (b); only the effective relationships are identified and then equally added in (c); the identified relationships are contribution-accordingly weighted and then added(d).

tation. Secondly, the relationship representations from the scene graph space are modified by spatially aligning into the visual space and redundancy elimination. Finally, the modified relationship representations are fed into an attention layer for representation weighting, which is the respective contribution quantification. Then, the concatenation of the weighted relationship representations and the image ones is fed into the depth predictor for depth estimation. Through this process, the relationships are spatialized for enhancing the depth estimation, as shown in Fig. 3.2.

Notably, recent advancements of depth estimation have also made significant progress in areas related to my work, since the publication of my contributions in this chapter was in 2022. Specifically, Duan et al. [55, 121] reformulate monocular depth estimation as a denoising diffusion process, which achieves great performances on both indoor and outdoor datasets. Xu [313] proposed to unify the different tasks including optical flow, stereo matching, and depth estimation. The unified framework adopt transformer architecture to learn discriminative features. Kong et al. [128] propose to learn depth estimation on corrupted datasets, which achieve robust estimation against different light conditions.

3.2 Method

In this section, we explain the proposed framework in detail. As depicted in Fig. 3.3, our framework comprises three stages. In Stage 1, the image representations of the input image are extracted from a deep learning-based MDE model. Meanwhile, a detection module is utilized on the input image to detect the objects in the scene, which constructs the scene graph. Afterward, the relationship representations are extracted from the scene graph by implementing a relationship recognition module. It is noted that here we adopt the same relationship categories as in Graph-rcnn [318]. In Stage 2, we first spatially align the relationship representations from the graph space with the image representations from the visual space. Then, the aligned relationship representations are modified by eliminating their redundancies compared with the image representations. In Stage 3, the concatenation of the modified relationship representation and the image representation is fed into the depth predictor the depth estimation, during which the attention is implemented on the modified relationship representations. The attention scores are automatically learned and used for identifying the effective relationships and quantifying their respective contributions on MDE.

3.2.1 Stage 1: Representation Extraction

Image Representation Extraction. The image representation extractor is implemented to obtain understanding from appearance cues, such as edges and shapes while eliminating the interference from surface textures. As shown in Fig.3.4, a U-Net encode-decoder network is adopted as our image representation extractor, where different resolution representations are combined by BiFPN[273] to preserve both the high-level understanding and visual details. Then, the image representation is:

$$R_I = f_{image}(I), \tag{3.1}$$

where I denotes the input image and f_{image} denotes the image representation extractor.

Relationship Representation Extraction. We first adopt an offline detector model [232] to detect the objects in the scene image so as to derive object-level understanding

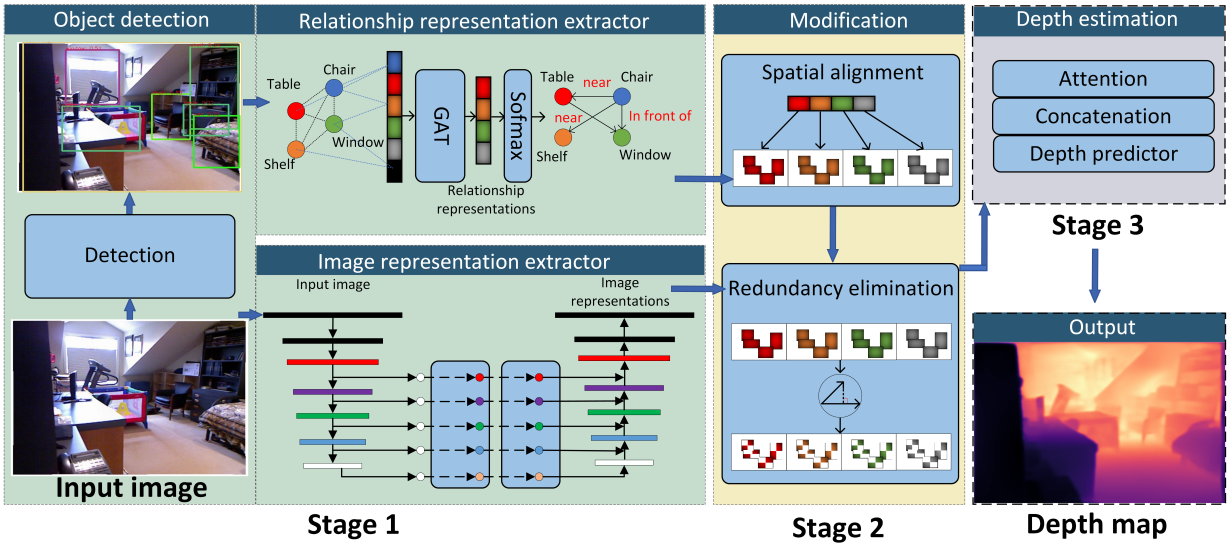


Figure 3.3: The framework of the proposed model. An input image is firstly sent to an image feature extractor to generate image representations. The input image is also fed to relationship recognition module to generate relationship representations. then, the relationship ones are aligned to the image ones and being eliminated with redundancy. Finally, the effective relationship representations are quantified and concatenated with image features to estimate depth.

and their spatial position:

$$z_i, M_i = Detector(I), i = 1, .., N, \tag{3.2}$$

where N denotes the number of the detected objects and z_i denotes the object-level understanding of the i-th detected object. M_i denotes the object mask, 1 for the object area. Then, the graph-rcnn [318] is adopted as the relationship recognition module. Specifically, the object-level understanding $(z_1, z_2, \cdots, z_N)$ of the N detected objects are treated as nodes features of the scene graph G. The $N \times N$ edges in the G correspond to $N \times N$ $< subject, object >$ relationships. Then, the node features are aggregated as:

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{i \in N(i)} a_{ij} W z_j^{(l)}\right), \tag{3.3}$$

$$a_{ij} = softmax(\sigma(W_a[z_i^{(l)}, z_j^{(l)}])), \tag{3.4}$$

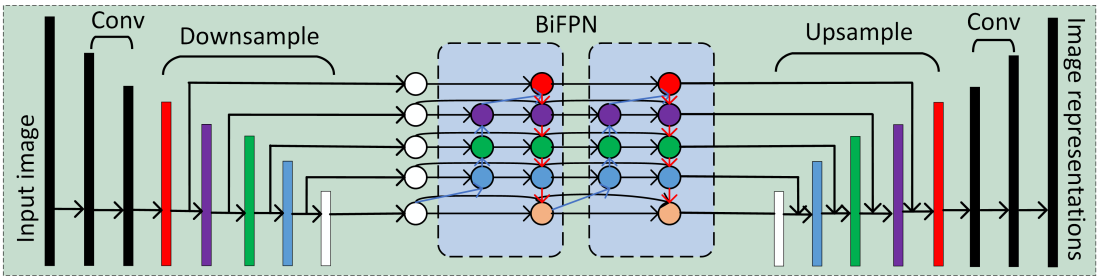


Figure 3.4: Image representation extractor

where $z_i^{(l)}$ denotes the i-th node at l-th iteration, σ denotes activation function, $N(i)$ denotes neighboring nodes of the i-th node, W and W_a are learnable parameters, and a_{ij} denotes attention between the i-th and j-th nodes. Afterward, the relationship representations are computed as:

$$R_r^{i,j} = embedding([z_i^{(l)}.z_j^{(l)}]). \tag{3.5}$$

Here the *embedding* is a fully-connected layer. Due to the deep learning-based depth estimation methods are only able to deal with fixed-sized input, we extract the fixed number of N_r relationship representations from the relationship recognition module. Specifically, we feed the relationship representations $R_r^{i,j}$ into a relationship predictor and preserve the ones with top-N_r prediction scores.

3.2.2 Stage 2: Representation Modification

After obtaining the image representations R_I and relationship representations R_r^n, where $n \in [1, N_r]$, we modify the relationship ones according to the image ones by first spatial alignment and followed by the redundancy elimination.

Spatial Alignment. Since the image representations, R_I are in the visual space and the relationship representations R_r^n are in the graph space, spatial alignment is implemented to translate the relationship ones into the visual space for depth estimation. Specifically, given the relationship representation $R_r^n \in \mathbb{R}^d$, where d denotes the representation dimension of R_r^n, and its corresponding object masks M_i and M_j, we first project the relationship

representation upon a zero-initialized feature set:

$$F^n(x, y, :) = R_r^n(M_i(x, y)||M_j(x, y)), x = [1, ..., I_w], y = [1, ..., I_h], \tag{3.6}$$

where $F^n \in \mathbb{R}^{I_w \times I_h \times d}$, I_w and I_h denote the width and height of the input image I, and $||$ denotes the logic operation *or*. Then, a single output channel convolutional layer is implemented on the feature set F to obtain the visually aligned relationship representation:

$$VA_R^n = Conv(F^n), n = 1, ..., N_r, \tag{3.7}$$

where $VA_R^n \in \mathbb{R}^{I_w \times I_h}$ denotes the visually aligned single channel feature map of each relationship representation.

Redundancy Elimination. After obtaining the visually aligned feature maps of relationship representations, we eliminate their redundancy information according to the image representations, since the overlap information between them may lead to overlook the distinguished spatial information contained in the relationships. In detail, the overlaps between the relationship representation and the image representation are eliminated by:

$$M_R^c = G(M_R^{c-1}, R_I^{c-1}), c = 1, ..., N_I, \tag{3.8}$$

$$G(M_R^{c-1}, R_I^{c-1}) = M_R^{c-1} - \frac{< M_R^{c-1}, R_I^{c-1} >}{< R_I^{c-1}, R_I^{c-1} >} R_I^{c-1}, \tag{3.9}$$

$$M_R^0 = VA_R, \tag{3.10}$$

where M_R denotes the modified relationship representation, N_I denotes the number of feature maps in the image representation R_I and $< \cdot, \cdot >$ denotes inner product. At the end of this stage, we obtain the modified relationship representation M_R.

3.2.3 Stage 3: Depth Estimation and Contribution Quantification

Once the the modified relationship representations M_R are obtained, we feed them into an attention layer for learning to identify their effectiveness and quantify their respective contributions:

$$\alpha^n = Attn(M_R^n), n = 1, ..., N_r, \tag{3.11}$$

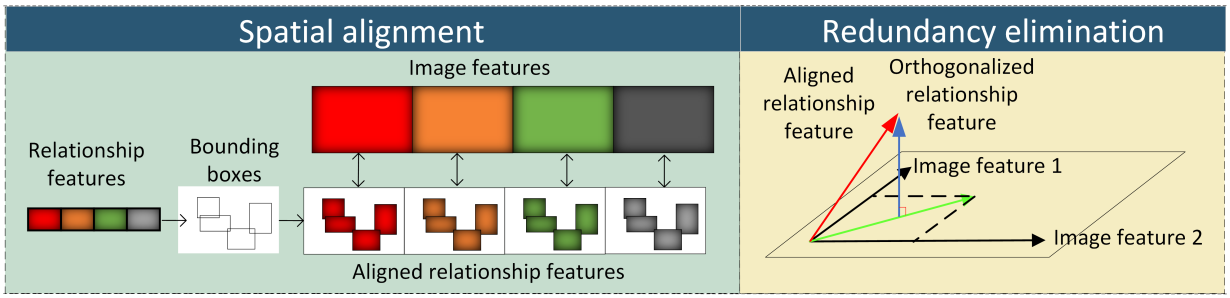


Figure 3.5: Representation modification is composed of spatial alignment and redundancy elimination. Firstly, relationship representations are spatially aligned towards image representations with bounding boxes. Secondly, the aligned relationship are orthogonalized against image representations to eliminate information redundancy.

where the attention scores α^n are used as the effectiveness identification and contribution quantification of the relationships. Then, we concatenate the image representations and the weighted relationship representations, feed it in the depth estimator for depth estimation:

$$y = Predictor(Concat(R_I, \alpha^n M_R^n)), n = 1, ..., N_r. \tag{3.12}$$

Then we impose the scale invariant loss [16], who focus on minimizing the ratio of the depth error, rather than the absolutely scaled depth:

$$Ł_{depth} = var([\log(y) - \log(\tilde{y})]) + \lambda[\frac{1}{N}\sum_{i=1}^{N}[\log(y) - \log(\tilde{y})]]^2, \tag{3.13}$$

where $var()$ denotes variance computation, $\tilde{y}$ denotes the ground-truth depth and λ denotes the balancing weight.

3.3 Experiments

In this section, we provide our experimental setups and show the results. Specifically, for the experiments, we first evaluate the performance of our framework on three MDE datasets. Then analysis the effectiveness of relationship spatialization, effective spatialized relationship identification and the respective contribution quantification are explained in detail. Finally, ablation study is conducted for measuring the necessary of the redundancy

elimination operation.

Our goal is, again, spatializing the inter-object relationships, then identifying the effective ones and quantifying their respective contributions on MDE. Since we are not aware of any existing work that performs same task, we aim to show the promise of the proposed framework, rather than trying to beat any state-of-the-art monocular depth estimation and relationship recognition models. More complicated networks, as long as they are end-to-end trainable, can be adopted to our framework to achieve potentially better results. More results are shown in supplementary materials.

3.3.1 Datasets and Implementation details

We adopt three datasets for depth estimation, KITTI [82], NYU Depth v2 [256], and ICL-NUIM [101] to validate the proposed framework. Since there is no dataset that provides both the ground-truth annotated depth and inter-object relationship, we evaluate our framework by the MDE performance. As for the relationship recognition, we pretrain a model on Visual Genome dataset[132], and adopt it directly in our framework. User studies are designed for evaluating the performance of the adopted relationship recognition module on the three depth estimation datasets.

KITTI[82]. It is a dataset of outdoor scene images with the corresponding 3D laser scans captured by moving vehicles. All the scene categories are used as our training/test sets. Following the split manner in [62], we use 26k images for training and 697 images for the test, with the image resolution of 1241×376. For a fair comparison, all compared MDE methods are retrained and evaluated on the adopted training/test sets in our experiments.

NYU Depth v2 [256]. It is a dataset of indoor scene images with the corresponding depth maps captured by Microsoft Kinect. Following the previous work, we use 50k images from 249 scenes for training and the 654-images set for the test, with the image resolution of 640×480. Different from the existing MDE methods like [16] that training only on cropped parts of the image, which may lead to the loss of relationship information, we perform the training on the entire image.

ICL-NUIM [101]. It is a smaller size dataset of an indoor scene, which comprises of

1500 images with their corresponding depth maps. Due to its limited amount of images, we adopt this dataset for verifying the generalizability of the model trained on NYU Depth v2.

Implementation. In the experiment, our framework is implemented with *Pytorch* [207] and with 4 NVIDIA P100 Pascal GPUs. We optimize with Adam [126], setting the learning rate $l = 5e - 4$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. Our framework is trained with batch size 16 for 25 epochs.

3.3.2 Performance Evaluation

In this part, we evaluate the performance of our proposed relationship spatialization framework on MDE. In order to show the effect of the learned relationship spatialization, we conduct the evaluations with our baseline framework and frameworks that implement the commonly used MDE networks as the image representation extractor. In the experiment, the state-of-the-art MDE methods, Adabins [16], Midas [225] and Bts [139], are adopted. We use Baseline, Adabins, Midas, and Bts to denote the MDE-only frameworks; and use Baseline+, Adabins+, Midas+ and Bts+ to denote the frameworks with relationship spatialization. For a fair comparison, all frameworks in the experiment are trained from the scratch.

KITTI. It can be seen from Table 3.1 that, for all chosen frameworks on KITTI dataset, relationship spatialization brings performance gain on almost every MDE evaluation metric, which shows that our proposed framework is able to learn to find the effective relationship spatialization for the outdoor scenes. Moreover, it is worth noting that the framework Bts+ performs obviously poor than Bts on the *silog* metric. Since the *silog* reflects the variation of scale error (calculated by dividing predicted depth with ground truth depth), the *rmse_log* reflects the mean value of the scale error, the BTS+ model focuses more on reducing mean error in relationship related areas and neglects areas without relationship enhancement, thus bringing in variation increment. So the BTS+ gets the greatest progress in *rmse_log* and does not outperform in *silog*.

NYU Depth v2. The results on NYU Depth v2 dataset are shown in Table 3.2, where the frameworks with the relationship spatialization outperform the original ones on almost

Metrics		Adabins	Adabins+	Midas	Midas+	Bts	Bts+	Baseline	Baseline +
↑	δ_1	0.9292	**0.9483**	0.9483	**0.9486**	0.8832	**0.9161**	0.9113	**0.9260**
	δ_2	0.9883	**0.9911**	0.9912	**0.9927**	0.9746	**0.9837**	0.9852	**0.9884**
	δ_3	0.9973	**0.9981**	0.9980	**0.9984**	0.9937	**0.9960**	0.9972	**0.9977**
↓	*abs_rel*	0.0775	**0.0758**	0.0655	**0.0652**	0.1195	**0.0831**	0.0917	**0.0789**
	rmse	3.4291	**2.9441**	**2.8099**	2.8747	4.4126	**3.4930**	3.6264	**3.2560**
	log_10	0.0348	**0.0339**	0.0291	**0.0289**	0.0486	**0.0368**	0.0399	**0.0342**
	rmse_log	0.1196	**0.1104**	0.1026	**0.1025**	0.1567	**0.1297**	0.1327	**0.1191**
	silog	10.6605	**9.5903**	9.3816	**9.3112**	**11.8858**	13.5749	12.3885	**11.0609**
	sq_rel	0.3406	**0.2875**	0.2660	**0.2603**	0.4026	**0.4026**	0.4042	**0.3554**

Table 3.1: Results of monocular depth estimation on KITTI dataset. ↑: the higher is better. ↓: the lower is better.

Metrics	Adabins	Adabins+	Midas	Midas+	Bts	Bts+	Baseline	Baseline+
δ_1	0.7971	**0.8603**	0.8173	**0.8519**	0.6936	**0.7715**	0.8293	**0.8889**
δ_2	0.9486	**0.9704**	0.9519	**0.9655**	0.9214	**0.9493**	0.9522	**0.9597**
δ_3	0.9840	**0.9928**	0.9858	**0.9901**	0.9803	**0.9852**	0.9791	**0.9894**
abs_rel	0.1537	**0.1203**	0.1437	**0.1282**	0.1772	**0.1557**	0.1488	**0.1432**
rmse	0.5318	**0.4469**	0.5016	**0.4717**	0.6652	**0.5847**	0.7632	**0.5342**
log_10	0.0629	**0.0515**	0.0596	**0.0544**	0.0806	**0.0703**	**0.0596**	0.0646
rmse_log	0.1900	**0.1588**	0.1822	**0.1779**	0.2327	**0.2245**	0.2015	**0.1924**
silog	15.9615	**13.9534**	15.4977	**15.4224**	**18.5223**	21.9191	17.9495	**15.8711**
sq_rel	0.1234	**0.0803**	0.1116	**0.0896**	0.1542	**0.1233**	0.1979	**0.1032**

Table 3.2: Results of monocular depth estimation on NYU Depth v2 dataset.

Metrics	Adabins	Adabins+	Midas	Midas+	Bts	Bts+	Baseline	Baseline +
δ_1	0.9160	**0.9521**	**0.8432**	0.7903	0.6735	**0.7758**	0.7344	**0.8446**
δ_2	0.9993	**1.0000**	0.9446	**0.9566**	0.9112	**0.9539**	0.9438	**0.9744**
δ_3	1.0000	**1.0000**	**1.0000**	0.9890	0.9783	**0.9882**	0.9902	**0.9943**
abs_rel	0.1048	**0.0758**	0.1878	**0.1521**	0.2075	**0.1634**	0.1732	**0.1297**
rmse	0.3283	**0.2584**	0.5462	**0.3492**	0.4727	**0.3726**	0.4158	**0.3141**
log_10	0.0438	**0.0319**	0.0761	**0.0637**	0.0843	**0.0694**	0.0709	**0.0545**
rmse_log	0.1276	**0.0959**	0.2059	**0.1936**	0.2417	**0.2373**	0.2076	**0.1681**
silog	12.6962	**8.5677**	20.8457	**19.2291**	24.0221	**23.6079**	20.5816	**16.7161**
sq_rel	0.0438	**0.0261**	0.1273	**0.0770**	0.1456	**0.0827**	0.1026	**0.0578**

Table 3.3: Results of frameworks trained on NYU Depth v2 dataset and evaluated on ICL_NUIM dataset.

all of the evaluation metrics. It shows that our proposed framework is able to learn to find the effective relationship spatialization of the indoor scenes. It is worth noting that, again, the Bts outperforms the Bts+ in the *silog* metric due to the same reason on KITTI.

ICL_NUIM. Due to its limited amount of training data, ICL_NUIM dataset is utilized for measuring the generalizability of our proposed framework. Specifically, we pre-train all frameworks on NYU Depth v2 dataset and evaluate them on ICL_NUIM dataset. As can be seen from Table 3.3, the frameworks with relationship spatialization outperform the original ones on all evaluation metrics, which shows the promising generalizability of our framework.

3.3.3 Effect of Relationship Representation

In this part, we evaluate the effect of relationship representations on MDE by performance comparison, correlation evaluation, related area evaluation, and framework derivation visualization.

Performance Comparison. To illustrate the effect of relationship representation for depth estimation, here we compare the performance of framework with only image representation and framework with both the image and relationship one. Note that, the attention layer in the predictor of our proposed framework is not implemented in this part, which means that all the relationship representations are equally treated. As shown in Table 3.4, the framework with relationship representations outperforms the original one, which shows the relationship representations benefit the depth estimation.

Correlation Evaluation. In this experiment, we evaluate the correlation between the MDE performance and the relationship recognition accuracy. Specifically, we adopt relationship recognition modules with different pre-trained accuracy into our framework and then observe the difference in the MDE performance. As can be seen from Table 3.5, we MDE performance increase with the pre-trained relationship recognition accuracy, which shows that correctly predicted relationships benefit the depth estimation.

Moreover, since there are no ground-truth annotated relationships in the MDE datasets, a subjective user study is designed for evaluating the accuracy of the pre-trained relation-

Models		δ_1	δ_2	δ_3	abs_rel	$rmse$	log_10	$rmse_log$	$silog$	sq_rel
KITTI	B	0.9292	0.9883	0.9973	0.0775	3.4291	0.0348	0.1196	10.6605	0.3406
	+R	**0.9446**	**0.9910**	**0.9981**	**0.0738**	**2.9988**	**0.0331**	**0.1098**	**9.5838**	**0.2892**
NYU	B	0.8237	0.9597	0.9895	0.1390	0.5006	0.0581	0.1770	15.0716	0.1042
	+R	**0.8608**	**0.9715**	**0.9932**	**0.1295**	**0.4487**	**0.0519**	**0.1576**	**13.2523**	**0.0796**

Table 3.4: The model added with relationship representations (+R) outperforms baseline model only using image representations (B) both on KITTI and NYU datasets.

Relationship			Depth estimation							
Recall	User_Acc		Adabins+		Midas+		Bts+		Baseline+	
	KITTI	NYU	KITTI	NYU	KITTI	NYU	KITTI	NYU	KITTI	NYU
10.5	79.62	78.58	2.9988	0.5318	3.0122	0.5134	7.8685	0.7915	4.5427	0.7463
12.5	81.27	83.01	2.9441	0.4654	2.9671	0.5016	7.5241	0.6246	4.1325	0.6824
16.7	84.53	83.27	2.8803	0.4487	2.9035	0.4925	7.1373	0.5874	3.9776	0.6596
18.3	88.17	86.31	2.8742	0.4456	2.8747	0.4791	6.5481	0.5831	3.8212	0.5342
18.8	89.33	87.25	**2.8652**	**0.4392**	**2.8099**	**0.4717**	**6.0439**	**0.5058**	**3.2560**	**0.5209**

Table 3.5: The MDE performance increases with the relationship recognition accuracy. *Recall* is used as the evaluation metric of the pre-trained relationship recognition module, $User_Acc$ is used as the metric for user-oriented accuracy and $rmse$ is used as the metric for MDE.

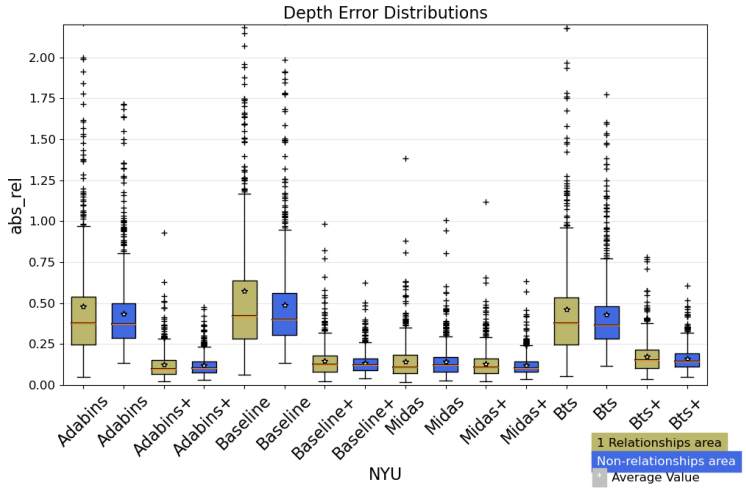


Figure 3.6: Distribution of depth errors on relationship-related areas and non-related areas for NYUv2 dataset. The yellow boxes denote the *abs_rel* on the relationship-related areas, the blue boxes denote the *abs_rel* on the non-related areas, the in-box red line denotes median value, and the star symbol denotes the mean value.

ship recognition module on the MDE datasets. Specifically, we involve 20 users and send each user 200 randomly selected relationships, which are used in our framework. Then, we ask the user whether each relationship is correctly recognized, according to which calculate the user-oriented relationship recognition accuracy. The user-oriented accuracy is close to the ground-truth one while covering only part of the used relationships. The user-oriented accuracy is shown in the second column in Table 3.5, it increases with the pre-trained accuracy.

Related Area Evaluation. Since the representations of the spatialized relationship are spatially aligned into the visual space, the majority of its spatial priors should have effects on the relationship-related areas. Although the performance on the entire image shows improvement, it is hard to fully reveal the effect of relationship spatialization for MDE. In this experiment, we compare the MDE performance on relationship-related areas and the non-related ones, the *abs_rel* metric is chosen for clear visualization. As can be seen from the box plots in Figure. 3.6, the introduction of the relationship, firstly,

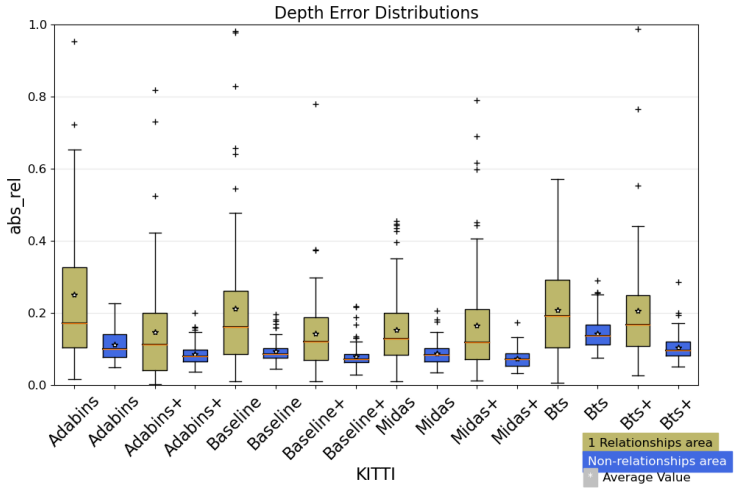


Figure 3.7: Distribution of depth errors on relationship-related areas and non-related areas for KiTTI dataset.

improves both the MDE performance on the relationship-related areas and non-related areas. More importantly, the overall depth error reduction on the relationship-related areas highly outperforms that on the non-related areas, which shows the large benefits from relationship spatialization on MDE.

Framework Derivation Visualization. In this experiment, we calculate and show the framework derivation map in Figure. 3.8. It can be seen that, after introducing the relationship representations, the derivation values are larger on the relationship areas than that on other areas, which means the relationship areas contribute more to depth estimation, thus more important. It is worth noting that, excluding the relationship areas, the sudden texture changing areas are with the highest derivation scores, which shows the fact that image representation-based methods abstract the spatial information from edge areas. Then, the introduction of our relationship spatialization helps to extract the spatial information from object surface areas, complement with the image representation-based MDE methods, thus improving the MDE performance.

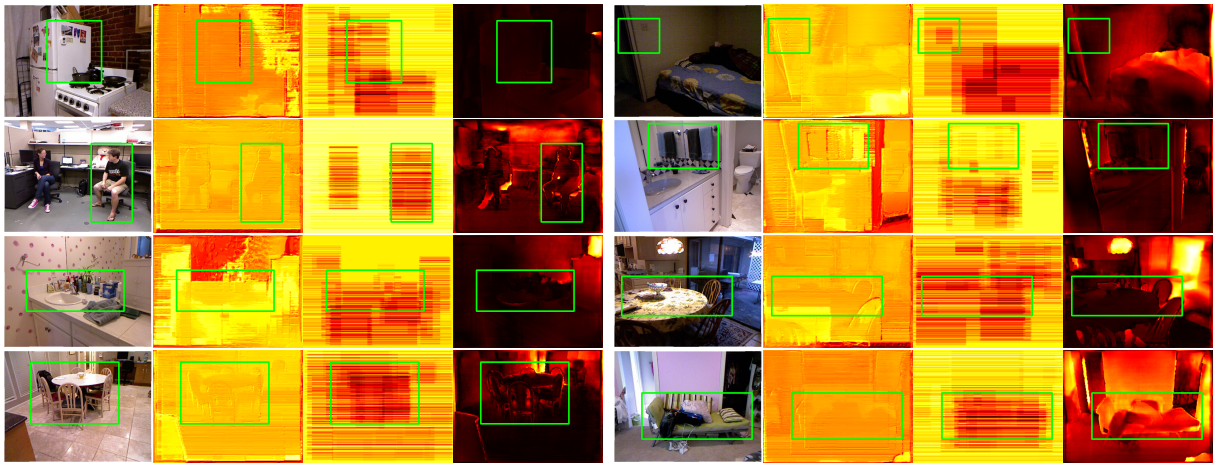


Figure 3.8: Framework derivation visualization. The first column of each group shows the input scene image, the second column shows the framework derivation map, the third column shows the relationship representation map, the fourth column shows the depth error map, and the bounding box denotes the relationship area.

3.3.4 Effective Spatialized-Relationship Identification

In this part, we evaluate the identified effective relationships from our framework by performance comparison and user study.

Performance Evaluation. In order to evaluate whether the identified effective relationships are meaningful for MDE, here we compare the performance of the framework with all relationship representations and the one with only the identified effective relationship representations. Specifically, a binary feature channel mask is learned from the relationship representations and used for effective relationship representations identification. Then, all identified effective relationship representations are equally treated to concatenate with the image ones for depth estimation. As shown in Table 3.6, the framework with only effective relationship representations outperforms the one with all relationship representations, which shows that the effective relationships for MDE are preserved and the noisy information from the non-effective relationships is eliminated.

User Study. Due to the lack of ground-truth annotations of the effective relationships in MDE, a user study is designed and performed to evaluate the identified effective

Models		δ_1	δ_2	δ_3	*abs_rel*	*rmse*	*log_10*	*rmse_log*	*silog*	*sq_rel*
KITTI	+R	0.9446	0.9910	0.9981	0.0738	2.9988	0.0331	0.1098	9.5838	0.2892
	+ER	**0.9512**	**0.9913**	**0.9981**	**0.0725**	**2.8574**	**0.0215**	**0.0905**	**9.4273**	**0.2138**
NYU	+R	0.8608	0.9715	0.9932	0.1295	0.4487	0.0519	0.1576	13.2523	0.0796
	+ER	**0.8752**	**0.9833**	**0.9932**	**0.1107**	**0.3583**	**0.0462**	**0.1113**	**12.4264**	**0.0686**

Table 3.6: The model added with effective relationship representations (+ER) outperforms the model with equally treated relationship representations (+R) both on KITTI and NYU datasets.

Models		δ_1	δ_2	δ_3	*abs_rel*	*rmse*	*log_10*	*rmse_log*	*silog*	*sq_rel*
KITTI	+ER	0.9512	0.9913	0.9981	0.0725	2.8574	0.0215	0.0905	9.4273	0.2138
	+QR	**0.9549**	**0.9914**	**0.9981**	**0.0690**	**2.8541**	**0.0208**	**0.0856**	**9.3253**	**0.1838**
NYU	+ER	0.8752	0.9833	0.9932	0.1107	0.3583	0.0462	0.1113	12.4264	0.0686
	+QR	**0.8820**	**0.9833**	**0.9932**	**0.1071**	**0.3552**	**0.0413**	**0.1004**	**12.1541**	**0.0666**

Table 3.7: The model added with quantified relationship representations (+QR) outperforms the model only using effective relationship representations (+ER) both on KITTI and NYU datasets.

Models		δ_1	δ_2	δ_3	*abs_rel*	*rmse*	*log_10*	*rmse_log*	*silog*	*sq_rel*
KITTI	B	0.9292	0.9883	0.9973	0.0775	3.4291	0.0348	0.1196	10.6605	0.3406
	+RE	**0.9433**	**0.9884**	**0.9973**	**0.0640**	**3.2055**	**0.0338**	**0.0982**	**9.5324**	**0.2760**
NYU	B	0.8237	0.9597	0.9895	0.1390	0.5006	0.0581	0.1770	15.0716	0.1042
	+RE	**0.8634**	**0.9722**	**0.9933**	**0.1205**	**0.4383**	**0.0509**	**0.1556**	**13.1205**	**0.0790**

Table 3.8: The model added with redundancy elimination (+RE) outperforms the baseline model only using image representations (B) both on KITTI and NYU datasets.

relationships from our framework. Specifically, we involve 107 users and send each user 100 randomly selected relationship sets shown in Fig.3.9. Each relationship set includes the input image and 4 images that each randomly gets rid of one of our identified effective relationships, where one of the two related objects is erased with an inpainting model [337]. Then, we ask the user if the erased relationship affects the measurement of the depth of the remaining object, then evaluate if we correctly identify the effective relationships. Our identified effective relationships finally achieve 89.33% accuracy, which is highly aligned with human recognition.

User study for respective contributions quantification Due to the lack of ground-truth annotations of the respective contributions, a user study is designed and performed to evaluate the quantified respective contributions from our framework. Specifically, we involve 107 users and involve each user in 100 multi-round tests. As shown in Figure. 3.10, for the first round of each multi-round test we first locate a target object in the scene, table in this case, and then erase one of its identified effective relationships Figure. 3.10(b-d), finally, we show the original scene image and its erased ones to the user. At the end of this round, the user is asked to choose the most important relationship for estimating the target object. Then in the next round, the erased image from the previous round that erases the relationship with the highest respective contribution is used as the ground-truth image. Afterward, we further erase one of the remaining effective relationships, Figure. 3.10(e-g), and ask the user to choose the highest contributed relationship in this situation. Finally, we evaluate the accuracy of our respective contributions according to the rank of the users. As can be seen from Table 3.9, for target objects, our top-5 accuracy is around 91.25%, which shows that our learned respective contribution is highly aligned with the human recognition.

cMetrics(%)	Top 1 Acc	Top 2 Acc	Top 3 Acc	Top 4 Acc	Top 5 Acc
	86.78	87.23	89.20	90.44	91.25

Table 3.9: Quantification of relationship effectiveness.

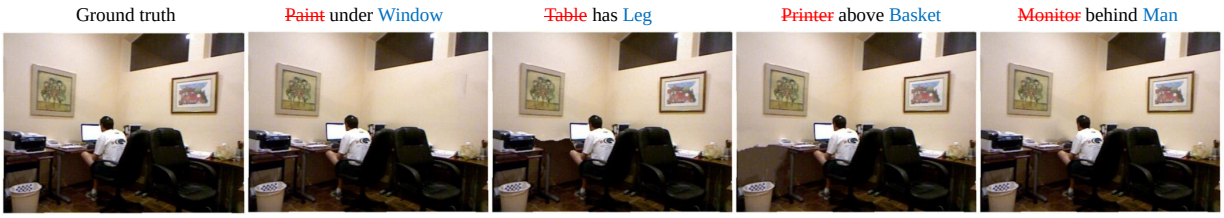


Figure 3.9: In each image, one target object of the identified effective relationships is erased. Then, users' judgments on whether the erased object is effective in depth estimation are recorded.

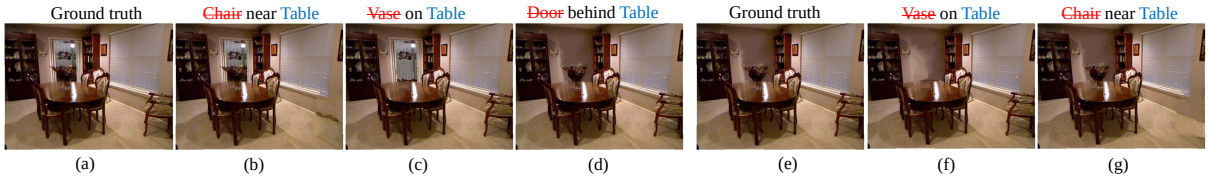


Figure 3.10: Quantification of relationships effectiveness. (a): ground truth image; (b)-(d): In the first round, erase each object and select the most important relationship; (e)-(g): In the second round, use image with the selected object erased as ground truth and select the next most important object.

3.3.5 Respective Contributions Quantification

In this part, we evaluate the quantified respective relationship contributions from our framework by performance comparison, correlation evaluation, and user study.

Performance Evaluation. To evaluate whether the quantified respective contributions from our framework are accurate, here we compare the performance of the framework that equally treats all relationship representations and the one that weights relationship representations according to their quantified respective contributions, which is our proposed framework. As shown in Table 3.7, the framework with quantified respective contributions outperforms the one with equal contributions, which shows that our respective contribution quantification spatialized the relationship adaptively.

Correlation Evaluation. In this experiment, we evaluate the correlation between the depth error and our quantified respective contribution to the relationship-related ar-

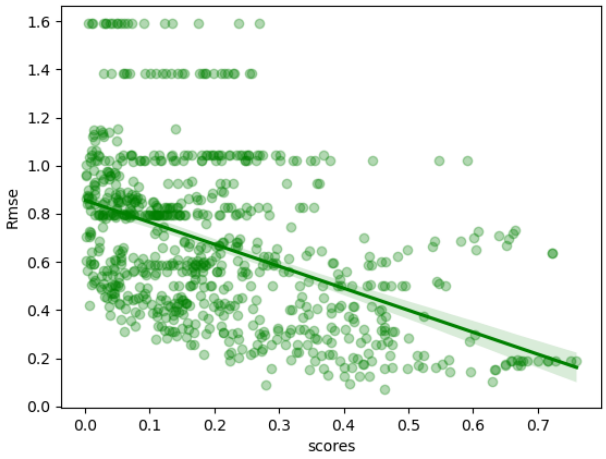


Figure 3.11: Correlation between the depth error and its quantified respective contribution on relationship areas.

eas. As can be seen from Fig. 3.11, the relationship area depth error decreases when its quantified respective contribution increases, which shows that our respective contribution quantification accurately ranks and spatialized the relationship for depth estimation.

3.3.6 Ablation Study

In this experiment, we verify the effectiveness of our redundancy elimination operation by comparing the framework with and without it. As can be seen from Table 5.6, the redundancy elimination operation leads to a marginal performance improvement on both the KITTI and NYU Depth v2 datasets, which is because it reduces the affection of redundancy information on the respective contribution quantification, thus increasing the focus on distinguished information from relationship spatialization.

3.3.7 Visualized results

In this section, we present the visualized results of the enhanced baseline model and state-of-the-art models against original baseline model and state-of-the-art models. The

test datasets include KITTI [82], NYU Depth v2 [256], ICL_NUIM [101] and visual-genome[132].

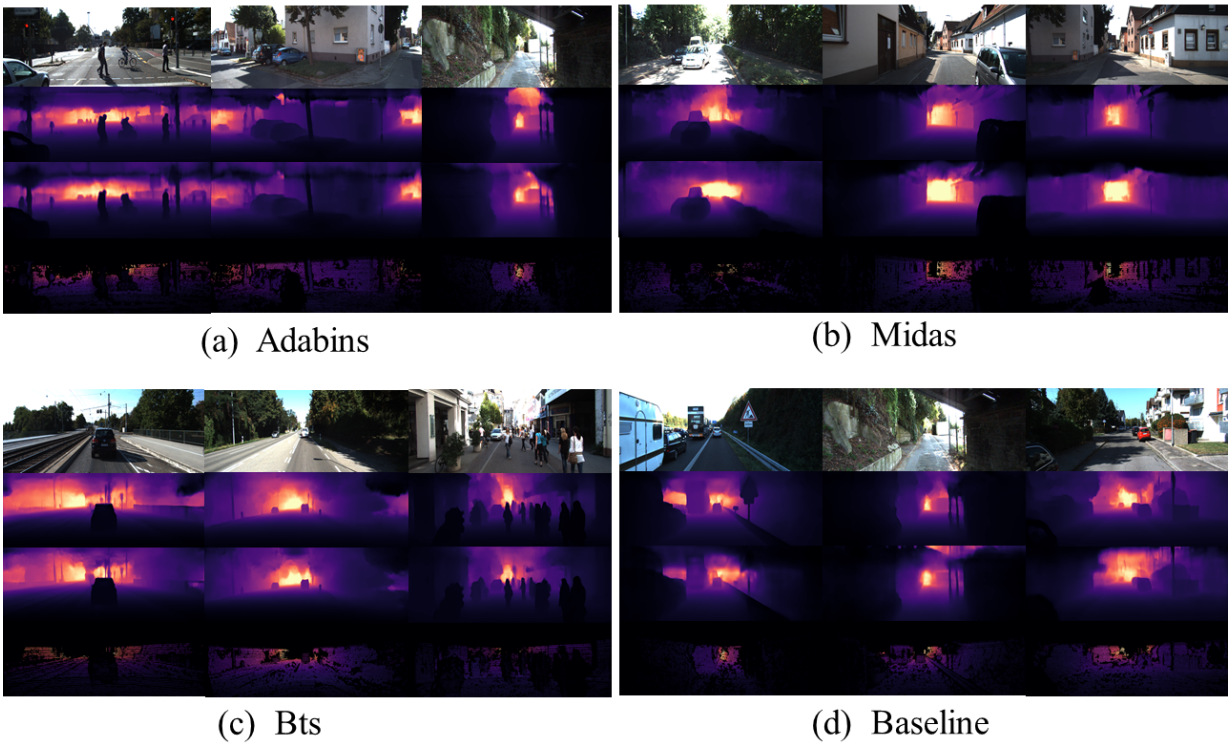


Figure 3.12: Visual examples on KITTI [82] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps.

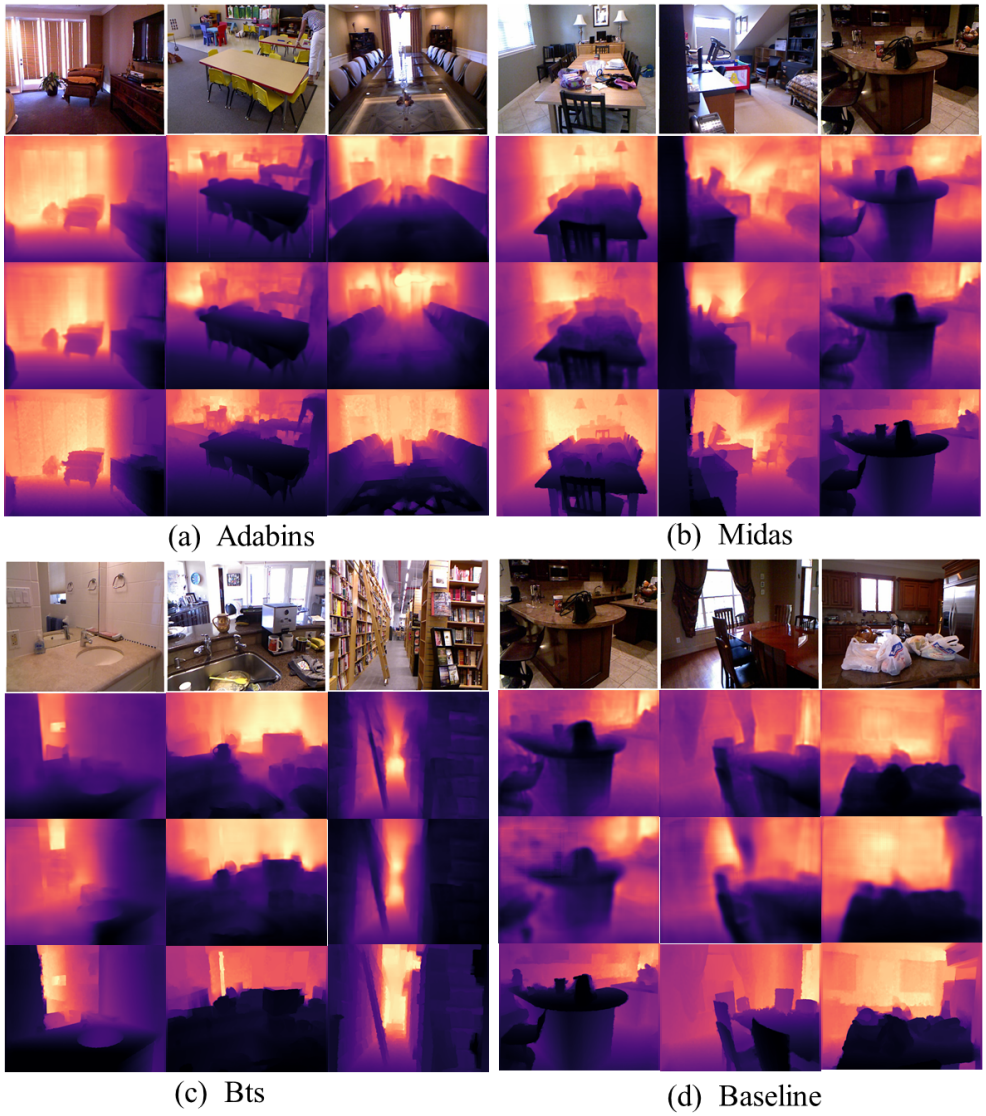


Figure 3.13: Visual examples on NYU Depth v2 [256] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps.

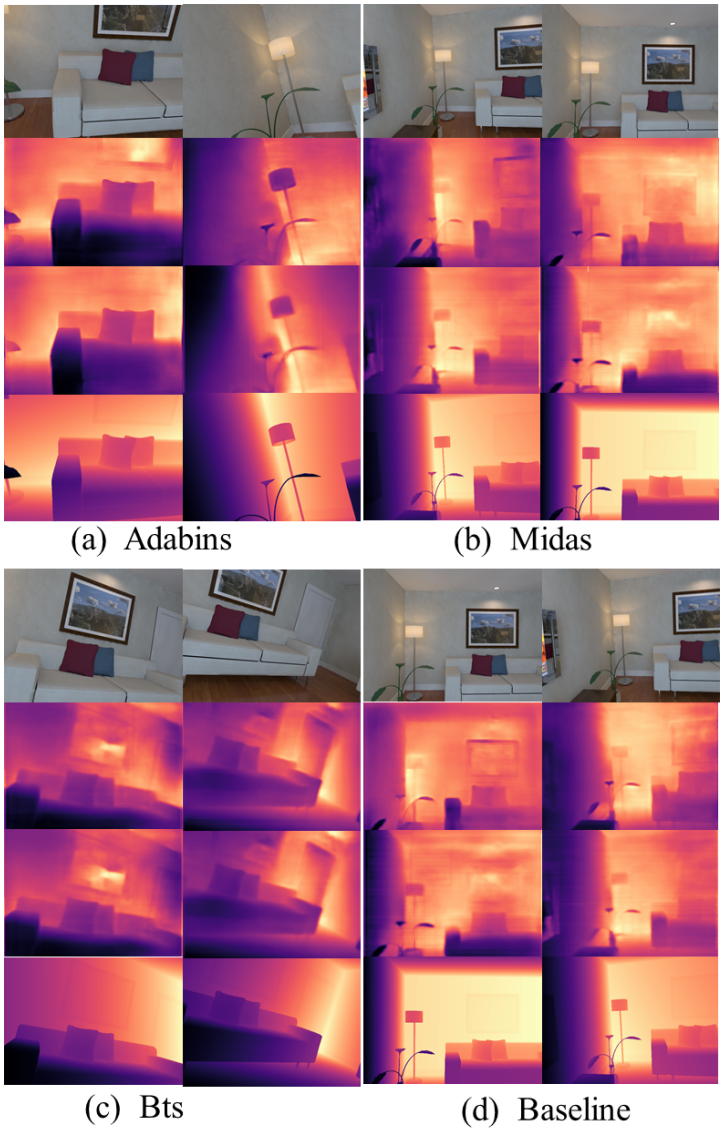


Figure 3.14: Visual examples on ICL_NUIM [101] test data. The first row: input images; The second row: relationships-enhanced depth maps; The third row: depth maps without relationships information; The last row: ground-truth depth maps.

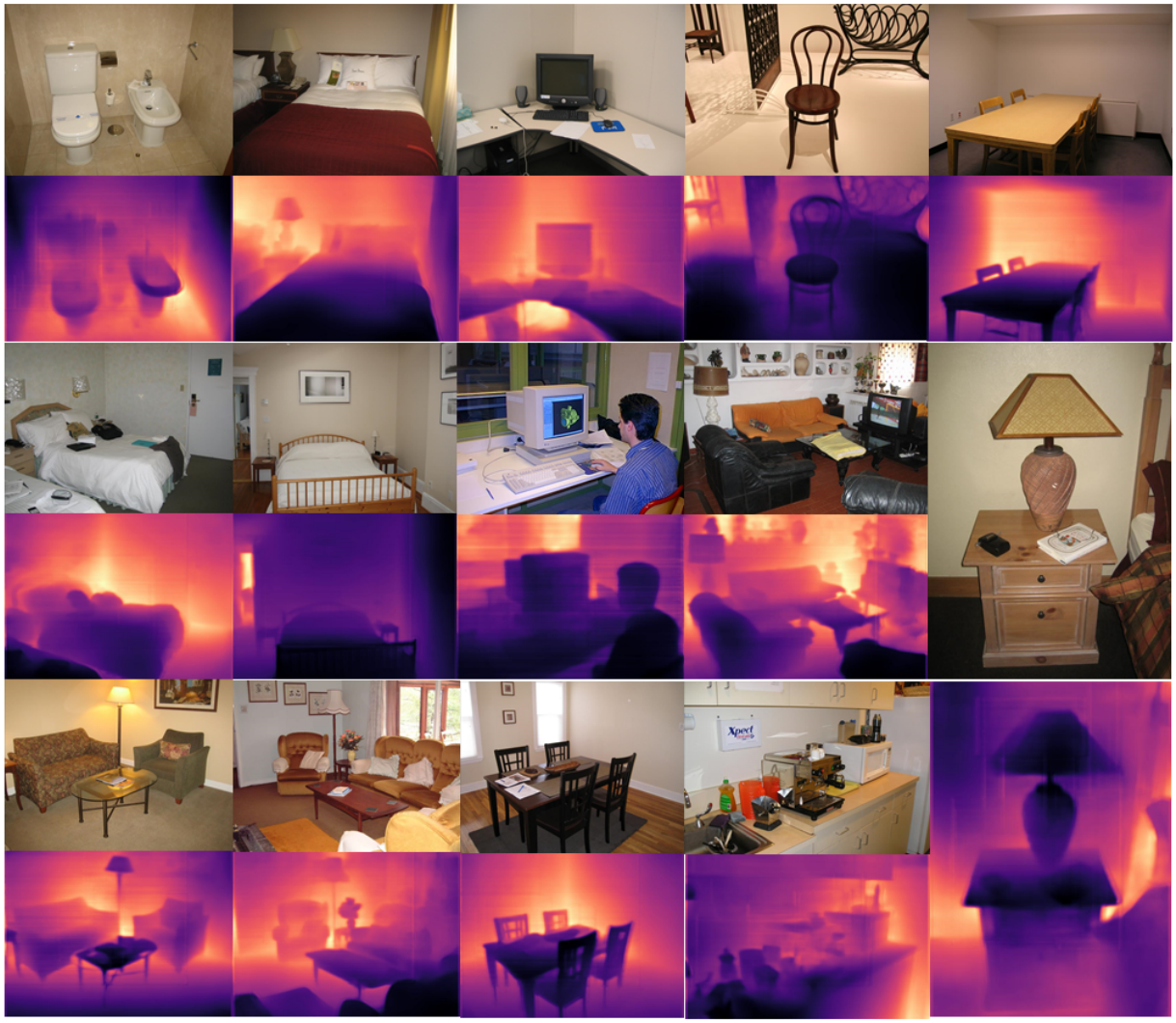


Figure 3.15: Visual examples on visual genome [132].

Chapter 4

Perceptual Relationship Transformation for 3D Understanding: 2D-images based novel view synthesis

4.1 Introduction

NVS aims to render new perspectives of a 3D model, a core task in computer graphics. Knowing that extracting inter-object relationships from 2D images enhances 3D understanding, we can leverage this approach to improve NVS. However, since the inter-object relationships we studied are based on monocular views, applying them to NVS requires relationship representations that hold across novel viewpoints. Predicting relationships for these novel views presents challenges, as relationship representations often lack robustness across varying perspectives. Looking at the visual relationship `(car-1, behind, car-2)` in Fig. 4.1(a) and asking to predict its corresponding ones in the novel views shown in Fig. 4.1(b-d), it can be found that the transformation of visual relationships is view-dependent and non-deterministic. The transformed relationships might be preserved, altered, or even become invisible in novel views. Therefore, given a single observed view

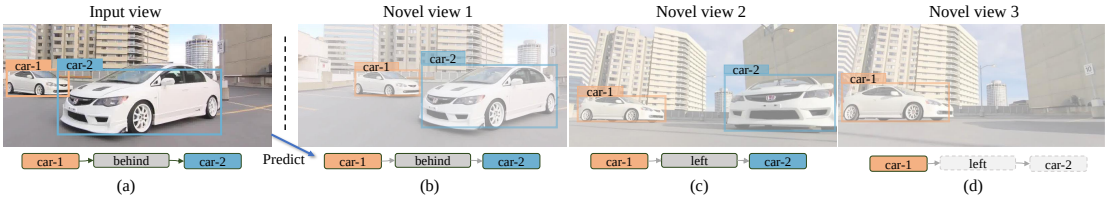


Figure 4.1: Visual Relationship Transformation. Given an observed view image (a), we aim to predict the relationships between objects (e.g., the one between `car-1` and `car-2`) in novel views (b-d) without requiring their corresponding view images. Compared with the relationship observed in (a), the corresponding ones in novel views (b-d) are typically view-dependent and non-deterministic, which might be preserved, altered, or even become invisible. The faded images and relationships denote unseen and invisible, respectively.

image and the cross-view transformations, can we learn to adaptively predict the transformed relationships and their presentness in novel views? We term this task, which has never been explored in prior works, as *visual relationship transformation* or VRT. The VRT plays a vital role in scenarios where a 3D understanding of novel views of a scene is demanded, particularly when the acquisition costs are expensive. For instance, when a robot is tasked with locating a referred object, it is necessary to understand the relationships between objects in the scene from various perspectives. Nonetheless, observing from every conceivable angle is expensive or impractical. So predicting the relationships in unseen views given observed views is a tractable and economical solution.

Comparing with numerous efforts in visual relationship detection (VRD) [132, 28, 213, 153, 148, 329, 351, 338, 251, 283, 216, 271, 367], VRT stands out by explicitly predicting the corresponding relationships in unseen views. In contrast, a vanilla solution, which first performs NVS and then conducts VRD, not only requires huge computations but also fails to maintain the correspondence between relationships from the source and target views. Furthermore, VRT comes with a large amount of 3D cues that help novel-view-related tasks. For example, the visual grounding task [3] shown in Fig. 4.2(a), which asks to identify the '`pillow on the far right`' from the unseen view of '`the front of the bed`', requires VRT for predicting the transformed relationships in the novel view, thus

can perform accurate alignment between the textual and visual semantics. Moreover, the novel-view-synthesis task [333], shown in Fig. 4.2(b), requires VRT to accurately predict the relationship transformation to the novel view, thus ensuring the correct relationship between the `potted plant` and the `bottle` in the synthesized image. Furthermore, the pedestrian intention estimation task [227], shown in Fig. 4.2(c), requires VRT to predict the relationships in the pedestrian view from the given camera view image, thus being able to estimate the pedestrian intention accurately.

While VRT can provide an informative understanding of novel views, learning to solve it remains a challenge. Firstly, labeling relationships in multi-view datasets is expensive, making large-scale training data unavailable. Secondly, in the novel view, the prediction of relationships and their corresponding presentness should be learned separately. As depicted in Fig. 4.1(d), the relationship should be predicted as `left` even though it is invisible in the novel view, where invisibility differs from unrelatedness. Thirdly, the predicted scene graph, constructed by the predicted relationships, should be structurally consistent with the ground-truth scene graph in the novel view, thus allowing the learning of global structure information.

Towards accomplishing VRT, we devise a novel approach to perform relationship transformation and tackle the aforementioned challenges during training. Specifically, given a single observed view image, we first detect the objects with an off-the-shelf detection algorithm. Each detected object is modeled as a node in the scene graph. This scene graph, along with the cross-view transformation, is then fed into the VRT module for predicting the transformed relationships, e.g. the edges in the scene graph, and their corresponding presentness. Throughout this process, the transformation equivariance of the relationship representations is enforced. To ensure the proper learning of relationship transformation, in the training stage, we first extract the ground-truth relationship features, presentness, and the scene graph from the novel view using a pre-trained visual relationship detection model [318]. Then, a relationship transformation loss is introduced to supervise each relationship prediction by adopting the ground-truth features as supervision, thus mitigating the lack of training data. Meanwhile, a presentness loss is implemented to learn the visibility of relationships in the novel view. Finally, a relationship structure loss is introduced to ensure the global structural information in the novel view can be learned. With these com-

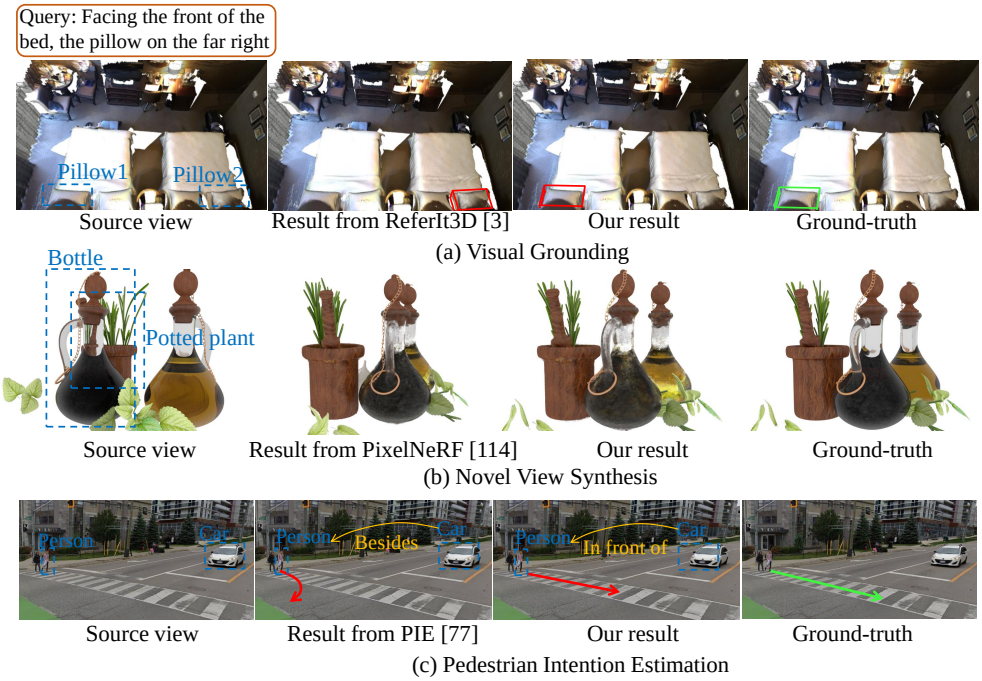


Figure 4.2: Illustration of VRT implementations on Visual Grounding, Novel-View Synthesis, and Pedestrian Intention Estimation tasks. For each task, the first image denotes the given input, the second one denotes the result from the SOTA method, the third one denotes the result by adopting our VRT as a plug-in module, and the fourth one denotes the ground-truth.

ponents, the trained approach is capable of performing visual relationship transformation in a structurally consistent manner.

Our contribution includes both a new VRT task and a novel approach designated to address the proposed task. To the best of our knowledge, this represents the first attempt in this direction. To evaluate the recognition accuracy of our VRT, we conducted extensive objective and subjective experiments and our VRT obtained encouraging results. Moreover, by adopting our VRT as a plug-in module, state-of-the-art methods in visual grounding [3], novel-view synthesis [298], few-shot NERF [333], and pedestrian intention estimation [227] have achieved considerable qualitative and quantitative performance improvement on Nr3D [3], Realestate10K [370], DTU [113], and PIE [227] datasets,

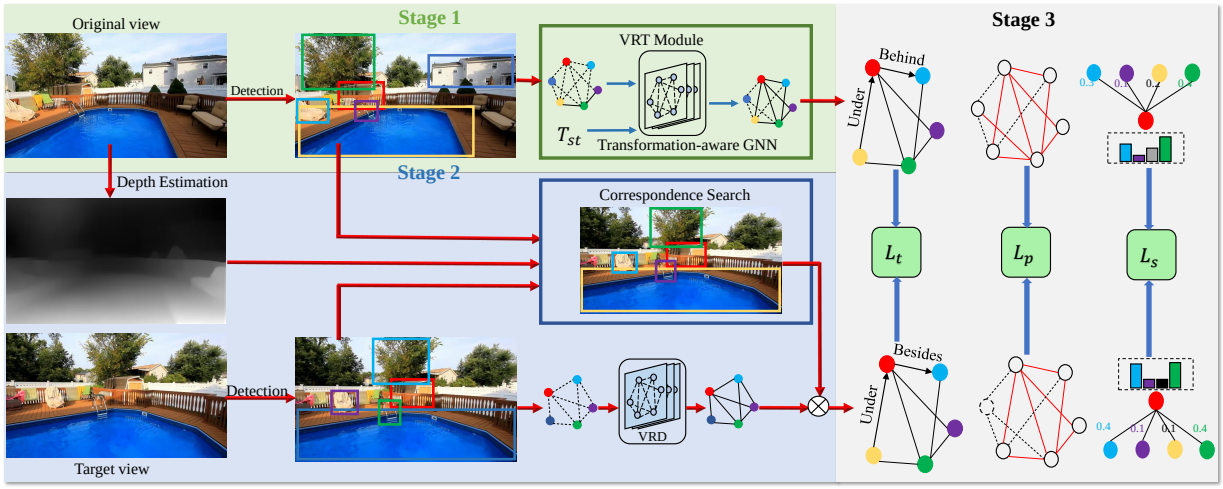


Figure 4.3: The illustration of the proposed approach. The T_{st} denotes the transformation matrix from the observed source view to the target view. $\bigotimes$ denotes the re-indexing and alignment operation. Note that, the relationship values and presentness are predicted by the VRT module, and the structure information is constructed by the predicted relationship and presentness.

respectively.

4.2 Method

In this section, we present the main VRT learning scheme. As illustrated in Fig. 6.2, our approach consists of three stages. In **Stage 1**, we initiate the process by detecting objects in the source view to construct a scene graph. This graph, coupled with the cross-view transformation, is then input into the VRT module for predicting transformed relationships and determining their presentness in the target view. In **Stage 2**, we search for matches between the predicted relationships, presentness, and the scene graph constructed accordingly, and their corresponding ground truth extracted from the target view. Finally, **Stage 3** introduces specific loss components: the relationship transformation loss for single relationship prediction, the presentness loss for predicting relationship visibility, and the relationship structure loss to preserve essential structural information.

4.2.1 Stage 1: VRT

In real-world scenarios, the observation of a scene from different views involves cross-view transformations, encompassing both rotation and translation, thus the observed relationships among objects in the scene should adhere to the spatial transformation. For example, in Fig. 4.1(b) and (c), the relationship between two cars changes from `behind` to `left` when observed from the side to the front. These two relationships are linked through the view transformation and conditioned on the scene. Therefore, we aim to predict the inter-object relationships in the target view while also keeping track of the changes in these relationships from the source to the target view. Motivated by the concept of equivariant neural networks, which preserves relationship correspondences over view transformations, an equivariant graph neural network is designed for learning transformation-equivariance representations of the relationships between objects.

Scene Graph Construction. Given an image from the original or source view, we initially employ an off-the-shelf object detector [232] to detect N objects and extract various pieces of information. This includes visual embeddings $\mathcal{X} = \{x_1, \ldots, x_N\}$, positional information in the form of bounding boxes $\mathcal{B} = \{b_1, \ldots, b_N\}$, semantic embeddings $\mathcal{W} = \{w_1, \ldots, w_N\}$ (i.e., word vector representing categories), and union region representations $\mathcal{U} = \{u_{1,1}, \ldots, u_{N,N}\}$. Subsequently, we construct a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the following manner. The node feature $v_i \in \mathcal{V}$ and the edge (or relationship) feature $e_{i,j} \in \mathcal{E}$ are defined as

$$\begin{aligned} v_i &= \phi_v(x_i \oplus \texttt{PE}(b_i) \oplus w_i), \\ e_{i,j} &= \phi_e(u_{i,j} \oplus v_i \oplus v_j), \end{aligned} \tag{4.1}$$

where $\oplus$ denotes the concatenation operation, $\texttt{PE}(\cdot)$ indicates the positional encoding operation [81]. ϕ_v and ϕ_e denote the equivariant layers [48], which firstly lifts the the size of the input from $N \times 1$ to $N \times 3$, and then perform representation learning with the equivariance-constraint:

$$\phi(\mathcal{T}'_g(x)) = \mathcal{T}_g(\phi(x)), \tag{4.2}$$

where $\mathcal{T}_g : X \to X$ is a set of transformations for the abstract group g.

Equivariant Message Passing. After the scene graph is constructed, we design an Equivariant Message Passing (EMP) module to model the correlations among nodes and

relationships while maintaining the equivariance property. Starting with the initial node and relationship features v_i^0 and $e_{i,j}^0$ from the previously constructed scene graph, the node message passing at step $l+1$ is formulated as follows:

$$v_i^{l+1} = v_i^l + \sigma(M_{i\cdot}^v + M_{\cdot i}^v), \tag{4.3}$$

$$M_{i\cdot}^v = \sum_{j \in \eta_i} \alpha_{ij} \phi_1(e_{ij}^l), M_{\cdot i}^v = \sum_{j \in \eta_i} \alpha_{ji} \phi_1(e_{ji}^l) \tag{4.4}$$

$$\alpha_{ij} = \frac{\exp(\sigma(\phi_2[v_i \oplus v_j]))}{\sum_{k \in \eta_i} \exp(\sigma(\phi_2[v_i \oplus v_k]))},$$

where σ refers to Vector-Neuron ReLU [48] and ϕ_1, ϕ_2 represent different Vector-Neuron Multi-layer Perceptron (VN-MLP) [48]. The term $M_{i\cdot}$ denotes the weighted averaged messages from the relationships between the i-th object and its neighbor objects η_i, while the $M_{\cdot i}$ denotes message from neighbors to the i-th object. Additionally, the formulation for relationship message passing is as follows:

$$e_{i,j}^{l+1} = e_{i,j}^l + \sigma(M_{i\cdot}^e + M_{j\cdot}^e), \tag{4.5}$$

$$M_{i\cdot}^e = \sum_{k \in \eta_i} \beta_{ik} (\phi_3([v_k \oplus \phi_4(b_i - b_k)])),$$

$$\beta_{ik} = \frac{\exp(\sigma(\phi_5[v_i \oplus v_k \oplus \phi_4(b_i - b_k)]))}{\sum_{l \in \eta_i} \exp(\sigma(\phi_5[v_i \oplus v_l \oplus \phi_4(b_i - b_l)]))},$$

where the $M_{i\cdot}^e$ and $M_{j\cdot}^e$ denote the aggregated message from the i-th and j-th objects to the relationship $e_{i,j}$. β_{ik} measures the contribution of message $M_{i,k}$ to the relationship $e_{i,j}$. ϕ_3 and ϕ_5 denote VN-MLPs, and ϕ_4 denots a VN-MLP with lifting operation [48]. The process of equivariant relationship message passing integrates relative position information into the relationship embedding while preserving equivariance, as the relative position is inherently equivariant to transformations such as rotation and translation.

Relationship Transformation. After the equivariant message passing, we predict the inter-object relationships in the target view. Let $\mathcal{E}_s = \{e_{i,j}\}$ denote the relationship representations from the source view V_s, for any target view V_t, we then predict its relationship representations $\hat{\mathcal{E}}_t = \{\hat{e}_{i,j}\}$ as follows:

$$\hat{\mathcal{E}}_t = \phi(T_{st} \cdot \mathcal{E}_s)^\top, \tag{4.6}$$

where ϕ represents an MLP layer, and T_{st} indicates the transformation matrix from the source view V_s to the target view V_t. In addition to the predicted relationship representations $\hat{\mathcal{E}}_t$, we also predict their presentness as follows:

$$\hat{p}_{i,j} = \texttt{sigmoid}\left(\texttt{MLP}\left(\left[v_i^l, v_j^l, e_{i,j}^l\right]\right)\right), \tag{4.7}$$

where l refers to the last message passing step, $\hat{p}_{i,j}$ indicates the probability of the relationship between node v_i and v_j being present in the target view. During inference, the presentness is determined as $\mathbf{1}\left\{\hat{p}_{i,j} \geq \gamma\right\}$, where γ is a threshold. In other words, it signifies that the relationship between the i-th and the j-th object is visible in the target view.

4.2.2 Stage 2: Correspondence Search

As described above, the VRT predicts the relationships and their corresponding presentness in the target view V_t based on the source view V_s and the inter-view transformation matrix T_{st} between the source and target views. To facilitate the training of VRT, it is necessary to align the predicted scene graph $\hat{\mathcal{G}}_t$ with the ground truth scene graph $\mathcal{G}_t$ in the target view V_t. To accomplish this alignment, we initially transform all object bounding boxes from the source view V_s to the target view V_t using the transformation matrix T_{st}. For each pixel (x, y) within a bounding box, we determine the transformed pixel coordinates $(\hat{x}, \hat{y})$ as follows:

$$[\hat{x}, \hat{y}, 1]^\top = K \cdot T_{st} \cdot D(x, y) \cdot K^{-1}[x, y, 1]^\top, \tag{4.8}$$

where $D(x, y)$ denotes the depth at pixel (x, y), obtained using a pre-trained depth estimation algorithm [236], and K represents the camera intrinsic matrix (i.e., both the source and target views use the same camera). With this setup, we can evaluate the overlap between each transformed bounding box and all bounding boxes in the target view. Specifically, if the transformed bounding box $\hat{b}_i$ has the maximum overlap with a target bounding box b_j among all bounding boxes in the target view, and $\hat{b}_i$ and b_j share the same object category label, we then identify a pair of nodes matched between the transformed scene graph and the target scene graph. For nodes and edges transformed from the source view that are invisible in the target view (i.e., fail to match with any others), we assign their ground truth presentness labels as 0, otherwise as 1.

4.2.3 Stage 3: Optimization

Through the identification of correspondences between the scene graph transformed from the source view and the ground truth scene graph in the target view, we train the proposed VRT model by aligning specific components of the two scene graphs using three different loss functions, as outlined below.

For every pair of the predicted (or transformed) relationship $\hat{e}_{i,j}$ and its corresponding ground truth relationship $e^*_{i,j}$, we define the **relationship transformation loss** as:

$$\mathcal{L}_t = \frac{1}{N_m} \sum_{i,j} \|\hat{e}_{i,j} - e^*_{i,j}\|_2^2, \tag{4.9}$$

where N_m denotes the number of matched relationship pairs, $e^*_{i,j}$ comes from the pre-trained Graph-RCNN [318] by taking the target view as input. For the prediction of relationship presentness $\hat{p}_{i,j}$, we define the **relationship presentness loss** as:

$$\mathcal{L}_p = \frac{1}{N_p} \sum_{i,j} \|\hat{p}_{i,j} - p_{i,j}\|_2^2, \tag{4.10}$$

where N_p represents the total number of edges (or relationships) in the scene graph constructed from the source view, and $p_{i,j}$ indicates the ground truth presentness label derived from the target view. In addition to relationship prediction, it is crucial to appropriately preserve the topology structure constructed by them. Inspired by [323], we characterize the topology structure of each edge $e_{i,j}$ through a set of its neighboring edges $\mathcal{N}(e_{i,j})$ that share one of the same nodes. Specifically, the information about the neighbor structure of edge $e_{i,j}$ is defined as a normalized distribution vector $\texttt{NS}_{i,j} \in \mathbb{R}^{|\mathcal{N}(e_{i,j})|}$. For each $e_{m,n} \in \mathcal{N}(e_{i,j})$, its corresponding element in $\texttt{NS}_{i,j}$ can be evaluated as:

$$\texttt{NS}_{i,j} = \frac{\exp\{||e_{i,j} - e_{m,n}||_2^2\}}{\sum_{e_{k,l} \in \mathcal{N}(e_{i,j})} \exp\{||e_{i,j} - e_{k,l}||_2^2\}}. \tag{4.11}$$

With the above definition of the neighbor structure, we then define the **relationship structure loss** as follows:

$$\mathcal{L}_s = \frac{1}{N_r} \sum_{(i,j)} \mathcal{D}_{\texttt{KL}}(\hat{\texttt{NS}}_{i,j} || \texttt{NS}_{i,j}), \tag{4.12}$$

where $\mathcal{D}_{\mathtt{KL}}$ denotes the Kullback-Leibler divergence, and $\hat{\mathtt{NS}}_{i,j}$ and $\mathtt{NS}_{i,j}$ represent the predicted and ground truth neighbor structure information, respectively. Finally, the overall loss function for training the proposed VRT is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_t + \lambda_2 \mathcal{L}_p + \lambda_3 \mathcal{L}_s, \tag{4.13}$$

where λ_1, λ_2, and λ_3 denote the weights used to balance different components.

4.3 Experiments

In this section, we present our experimental setups and results. Since no existing work performs exactly the same task as we do here, we mainly focus on showcasing the effectiveness of the proposed approach in two main aspects. Firstly, we compare the relationship prediction performance of our VRT approach with scene graph generation methods relying on visible target views. Additionally, through the adoption of VRT, we demonstrate its enhancements in various related tasks, such as novel view synthesis (leveraging Synsin [298] and PixelNeRF [334]), pedestrian intention prediction (based on [227]), and visual grounding (based on ReferIt3D [3]). It's important to note that our objective is to showcase the potential of learning visual relationship transformation to enhance multi-view understanding, rather than attempting to outperform all state-of-the-art methods.

4.3.1 Datasets and Implementation Details

We present the datasets used in our experiments as follows: For relationship prediction experiments, we employ the Realestate10K dataset [370] and assess generalizability using the VidVRD dataset [115]. Novel view synthesis experiments utilize both the Realestate10K dataset [370] and the DTU dataset [1]. Pedestrian intention estimation and trajectory prediction experiments rely on the PIE dataset [227]. Visual grounding experiments involve the Nr3D dataset [3]. Below, we provide a brief introduction to each dataset.

Realestate10K [370]. It contains 80K video clips from both indoor and outdoor scenes, comprising more than 10 million frames annotated with camera intrinsic parameters

and camera poses obtained from SfM. Following [298], we use 57K, 14K, and 7K scenes for training, validation, and testing, respectively. Notably, all scenes in the test set are unseen. We sample views by first selecting an original view and then choosing the target view with a maximum of 70 frames apart. Subsequently, we select frame pairs with an angle change of $\geq 5^\circ$ and a position change of ≥ 0.15. Finally, we obtained 16K, 4K, and 2K frame pairs for training, validation, and testing, respectively. In the novel view synthesis experiment, all selections are the same except for the inter-frame distance. The target views are chosen with a maximum of 30 frames apart from the original view, and we report the results on 2K samples from [298].

VidVRD [115]. It contains 1000 videos with 4835 annotated relationship instances spanning 132 categories. The training and validation sections consist of 800 and 200 videos, respectively. Due to the limited number of relationship annotations in VidVRD, we select a total of 500 image pairs from the validation videos to test the generalizability of VRT. The selected image pairs adhere to the same criteria for angle and position changes as in Realestate10K. The transformation matrices are generated using COLMAP [245].

DTU MVS [1].]. For each object, it contains 49 or 65 images, with camera information, baseline, and structured light ground-truth. The presented objects pose a challenge as their appearance changes in different viewpoints due to specularities. We adopt this dataset for novel view synthesis, following the training settings in [334].

PIE [227]. It consists of over 6 hours of video footage capturing pedestrians in various types of crosswalks, recorded by an onboard camera. It includes both ground-truth pedestrian intentions and trajectories, along with bounding boxes for traffic objects, pedestrian attributes, and camera parameters. There are 1,842 pedestrian samples and 293K annotated frames. Following [227], we use 150K, 30K, and 113K frames for train, validation, and test sets, respectively.

Nr3D [3]. It annotates 707 indoor 3D scenes with 45,503 human utterances. Follow the setting of ReferIt3D [3], we use 30K, 2K, and 8K image-query pairs for training, validation, and testing, respectively. Furthermore, during the testing, we report the results on both view-dependent and view-independent image-query pairs. **Implementation Details.** The proposed approach is implemented with Pytorch on four NVIDIA V100 GPUs.

In the training process, the batch size is set to 16. Adam optimizer is adopted with a fixed learning rate of 0.0001. The loss weights λ_1, λ_2, and λ_3 are set to be 0.5, 0.5, and 0.1. During inference, the threshold γ is set to be 0.5.

4.3.2 Visual Relationship Transformation

		SGCls		PredCls	
Method	Input View	R@50	R@100	R@50	R@100
VRT	Original	30.3	34.7	33.6	37.8
Synsin [298]+GRCNN [318]	Original	23.7	24.1	25.3	27.7
Synsin [298]+TDE [278]	Original	23.8	24.3	25.6	27.8
Synsin [298]+BGNN [146]	Original	24.0	24.4	25.2	28.0
GRCNN [318]	Target	31.0	34.9	34.0	38.2

Table 4.1: Results evaluated with Recall@50 and Recall@100 on tasks: relationship classification (PredCls) and scene graph classification (SGCls).

We conduct both objective and subjective evaluations on hand-labeled Realestate10K [370] datasets. Then, we conduct the generalizability test on the VidVRD dataset. Specifically, we classify transformed relationships using a pre-trained classifier [318]. Only the relationships predicted to be visible in the target view are considered for evaluation.

Objective Evaluation. Following the relationship categories from the Visual Genome dataset, on which the VRD is pre-trained, we manually label 500 image pairs from the test set of Realestate10K, encompassing a total of 10K relationships across 51 categories. These hand-labeled images are used only for testing purposes. Subsequently, we conduct a comparative analysis with 1) the VRD test on the synthesized view initially takes the original view image as input, synthesizes the novel view using Synsin [298], and then detects visual relationships on the synthesized view; 2) Graph-RCNN [318], TDE [278] and BGNN [146] tests are implemented on the target view, which is unavailable for our VRT model; and 3) Our VRT predicts the relationship in the target view based on the original view image. In Table 4.1, when provided with the same original views as input, the VRT

outperforms the VRD on the synthesized view. Furthermore, despite making predictions without the target view, our VRT achieves results comparable to Graph-RCNN [318]. The results also reveal that despite a better VRD model such as TDE [278] being given, the performance on synthesized novel views remains constrained. This underscores that the quality of the synthesized views is a significant bottleneck. Therefore, it proves that directly predicting transformed relationships from the original view is more effective and efficient.

Generalizability Evaluation. We further validate the generalizability of the VRT on VidVRD [115] dataset. Since the relationship categories from VidVRD are different from the pre-trained Graph-RCNN [318], we impose a relationship mapping from the visual-genome [132] annotations to the VidVRD [115] annotations, details of which can be found in supplementary materials. The VRDFormer [367] is tested with the target view as input, while our VRT only takes the original view as input. The results in Tab.4.2 show the VRT outperforms the VRDFormer [367] which demonstrates the VRT successfully learns multi-view understanding when only the original view is given.

4.3.3 Ablation Studies

Transformation Sensitivity. Given that the proposed approach is transformation-related, we conduct experiments here to test the sensitivity of our VRT to transformations. Specifically, we split the testing frame pairs into three sets: 1) T_1 with angles $\leq 10°$ & translations ≤ 0.20; 2) T_2 with angles $\leq 15°$ & translations ≤ 0.25; and 3) T_3 with angles $> 15°$ *or* translations > 0.25. In Tab. 4.13, we see that VRT performs well at different transformation levels, demonstrating that VRT is transformation-insensitive.

Method	Input View	R@50	R@100
VRDFormer [367]	Target	18.59	22.37
VRT	Original	21.35	25.41

Table 4.2: Results compared with VRDFormer [367].

Transformation Awareness. During the information aggregation of VRT, the equivariant GNN can involve the transformation of information in different stages. In this ex-

	Recall@50	Recall@100
T_1	31.5	35.8
T_2	30.6	34.2
T_3	29.8	33.6

Table 4.3: Results of VRT with different transformations.

	Recall@50	Recall@100
T in nodes	29.5	32.4
T in edges	29.4	32.2
T in attention	29.2	31.5
T in MP	30.3	34.7

Table 4.4: Results of involving the transformation information in different stages of information aggregation.

	Recall@50	Recall@100
VRT	30.3	34.7
VRT w label	22.2	23.6
VRT w/o $\mathcal{L}_p$	27.8	28.7
VRT w/o $\mathcal{L}_s$	25.5	27.0

Table 4.5: Results of ablation study.

periment, we conduct experiments on the hand labeling data to evaluate the performance of involving a transformation in only the nodes representation, only the edge representation, only the attention calculation, and the whole message-passing process. As shown in Tab. 4.14, the message-passing one performs the best. This is because the transformation equivariance should be preserved by the node representations, edge representations, and attention, which can only be effectively captured by the message-passing process.

Ablation Study. Here, we show the effectiveness of different loss terms. Specifically, in the first experiment, we replace the predicted and ground-truth representation with one-hot coding, where the predicted one-hot is obtained from `argmax` and the ground-truth one comes from the pre-trained Graph-RCNN [318]. Then, we conduct experiments without either the relationship presentness loss or the relationship structure loss. As shown in Tab. 4.5, after replacing the representations with one-hot labels, the performance decreases a lot. This is because, the same visual content can be annotated as different relationships, where the one-hot label breaks the similarity between different relationships. Furthermore, after removing the relationship structure loss, the performance decreases dramatically, which shows the importance of preserving the structural information.

4.3.4 Novel View Synthesis

We evaluate the effectiveness of VRT on novel view synthesis tasks. Instead of hard constraining the view synthesis by providing the one-hot relationship prediction, we utilize the relationship features from the pre-trained VRT model in the process of synthesizing, which is achieved by adopting the VRT model as a plug-in module. During which, the relationship features are spatially aligned into the image space by relationship spatialization operation in [314]. Specifically, two state-of-the-art novel view synthesis methods, Synsin [298] and PixelNeRF [333], are used for evaluation.

Synsin [298]. We evaluate the effectiveness of VRT on Synsin by adopting VRT as a plug-in module to provide relationship features (denoted as Synsin+), concatenating them with the global image features extracted by the sub-module of Synsin. Following [298], the model is retrained on Realestate10K [370]. As shown in Fig. 4.4, VRT helps preserve

	PSNR ↑	SSIM[296]↑	Perc Sim ↓
Synsin+	22.78	0.810	1.00
Synsin	22.31	0.740	1.18

Table 4.6: Results on Realestate10K [370] dataset. Synsin+ denotes enhanced with VRT



Figure 4.4: Visual results from Synsin [298] on Realestate10K [370]. The first two columns show the images from the original view and the target view, the third one shows the synthesized images from Synsin, and the fourth one shows the images from Synsin+ that adopt VRT as a plug-in module. The right four columns present that the synthesized images get better quality when relationships are predicted correctly. The dashed boxes denote invisible relationships in the target views.

the correct visual relationships in view synthesis. The quantitative results are shown in Tab. 4.6, where VRT brings considerable performance improvement on all metrics.

Upon observation, we note that Synsin may display insensitivity to large transformations, potentially resulting in structural inconsistencies. For example, in the seventh image of the third row in Fig. 4.4, given large the inter-view transformation, Synsin generates the inadequately transformed image, while the structural consistency is enhanced by VRT as shown in the third image, possibly because composing predicted relationships helps preserve the structural information during transformation, thus enhancing the consistency

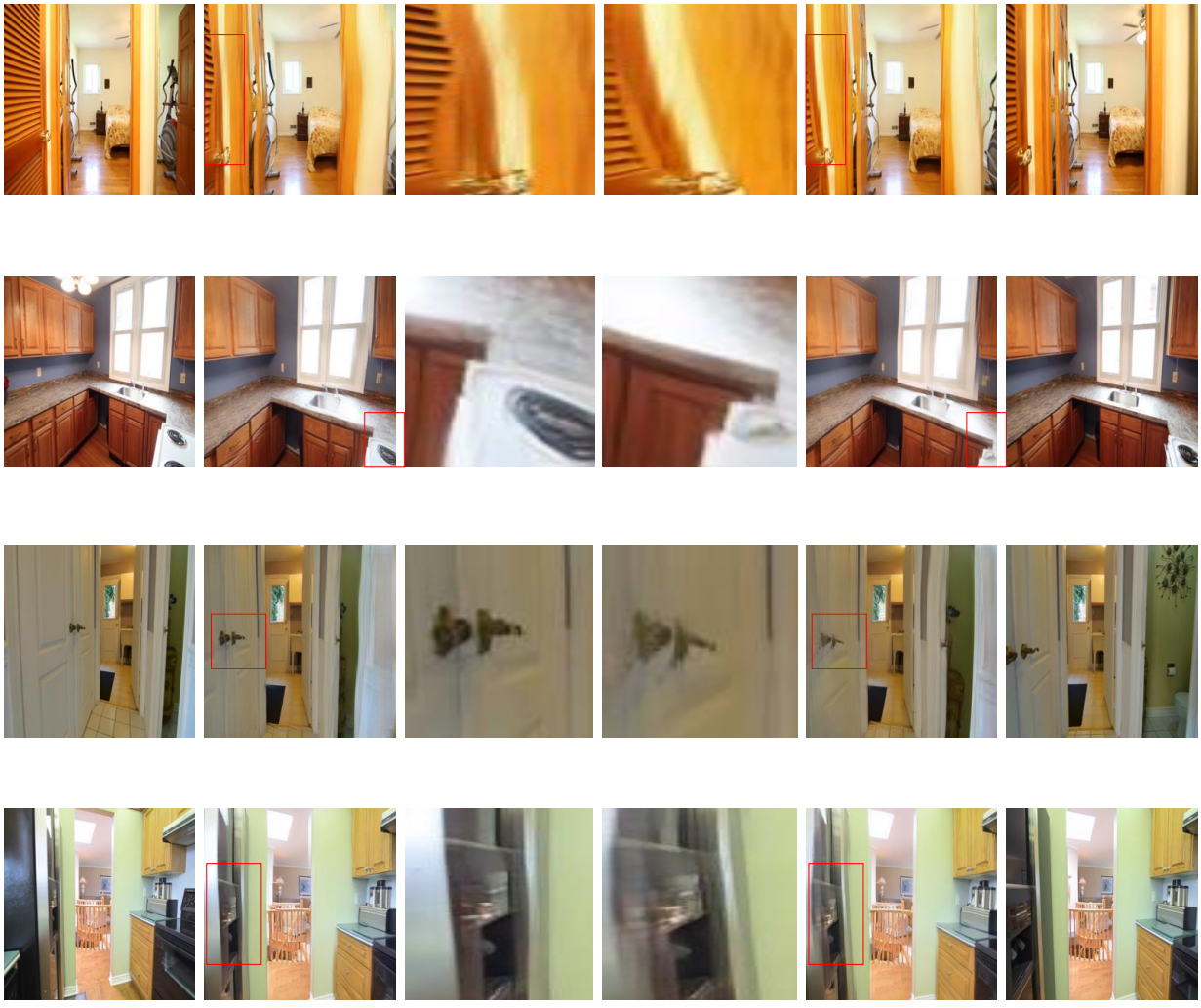


Figure 4.5: Visual results from Synsin [298] on Realestate10K [370]. The first column shows the input image, the second and the fifth columns show the output of enhanced and original Synsin, and the third and the fourth columns present the zoomed-in parts..



Figure 4.6: Visual results from PixelNeRF [333] on DTU MVS [1].

between the synthesized image and the target one. In Tab. 4.7, we follow the same setting in Sec. 4.3.3 to evaluate VRT on preserving structural consistency at different transformation scales, where Synsin+ becomes more robust to transformations.

PixelNeRF [333]. To evaluate VRT on PixelNeRF, we adopt it as a plug-in module to provide relationship features, which are then concatenated with the global image features extracted by the sub-module of PixelNeRF. Following the training setting in [333], the model is retrained on DTU MVS [1] dataset. The visual results are shown in Fig. 4.6, where three views are used for training PixelNeRF and we adopt one of them in VRT to predict the transformed relationships. As shown in Tab. 4.6, the model with VRT achieves considerable performance improvement on all metrics among all training settings.

4.3.5 Pedestrian Intention Estimation

We evaluate the effectiveness of VRT in estimating pedestrian crosswalk intention by adopting it as a plug-in module for [227]. Specifically, instead of utilizing car-view relationships, our VRT provides the feature of relationships in the pedestrian view, which is concatenated with the global image features extracted by the CNN module, and then fed into the LSTM module for intention estimation. Since it also benefits the prediction of pedestrian

		PSNR	SSIM[296]	Perc Sim
T_1	Synsin+	19.98	0.712	1.301
	Synsin	19.83	0.678	1.431
T_2	Synsin+	19.90	0.685	1.210
	Synsin	18.34	0.600	1.285
T_3	Synsin+	19.82	0.732	1.022
	Synsin	17.86	0.584	1.130

Table 4.7: Quantitative results on different transformation scales.

	PSNR	SSIM[296]	Perc Sim
PixelNeRF(1 v)+	15.91	0.618	0.465
PixelNeRF(1 v)	15.55	0.537	0.535
PixelNeRF(3 v)+	20.28	0.713	0.298
PixelNeRF(3 v)	19.333	0.695	0.387

Table 4.8: Results on DTU MVS [1]. 1v and 3v denote the number of available views in the training process.

	Overall	Videw-dep	View-indep
ReferIt3D [3]	0.356	0.325	0.371
ReferIt3D+	0.455	0.467	0.451

Table 4.9: Results on Nr3d [3] datasets.

	Acc ↑	F1 ↑	MSE ↓	C-MSE ↓
PIE+	0.8960	0.9446	551.29	513.23
PIE [227]	0.7905	0.8747	559.377	520.53

Table 4.10: Results on PIE [227].

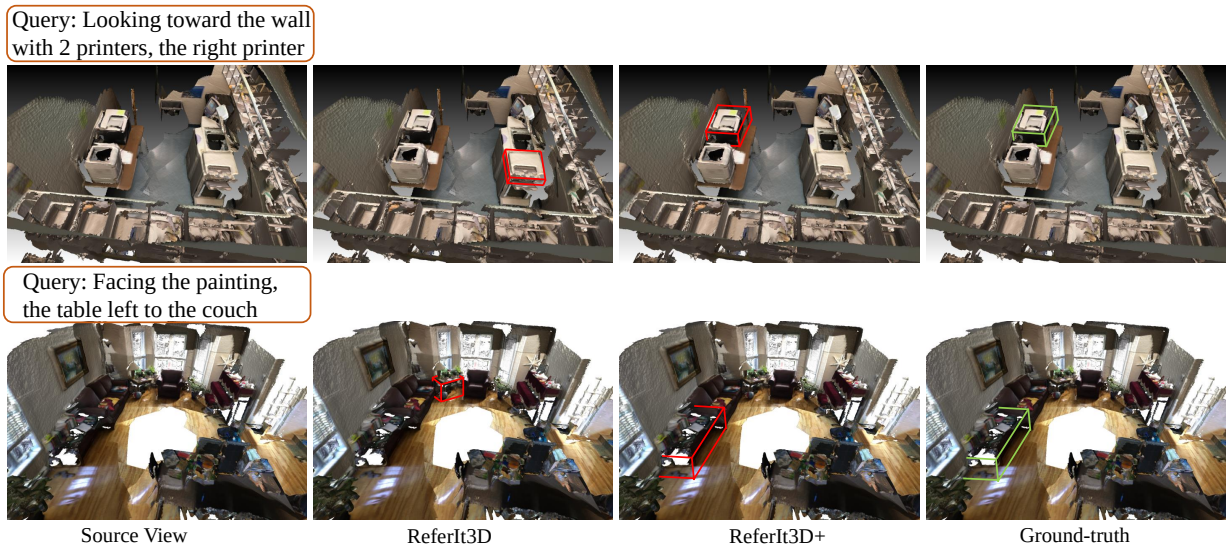


Figure 4.7: Visual results from RerferIt3D [3] and RerferIt3D+.

trajectory in a similar way, we thus conduct experiments on both pedestrian intention estimation and trajectory prediction. As shown in Tab. 4.10, both two tasks gain significant performance improvements. This is because humans perform crosswalks and trajectory planning according to the inter-object relationships in their view, rather than the view of the car.

4.3.6 Visual Grounding

We evaluate VRT for visual grounding by integrating it with ReferIt3D [3]. As shown in Fig. 4.7, RerferIt3D+ enhanced by VRT can locate more accurately than ReferIt3D [3]. Furthermore, we train and test ReferIt3D [3] enhanced by VRT, and Tab. 4.9 shows the improvements on location accuracy, where the improvement on view-dependent queries is larger than the view-independent. This demonstrates that VRT endows ReferIt3D [3] with multi-view understanding.

4.3.7 Results on KITTI

KITTI [83] contains 18560 consecutive training images and 4641 images for testing in complex city scenes. To create frame pairs for training and testing, we sample two images at once if they lay within 10 frames away. For the rotation and translation constraints, we adopt the same strategy as in Realestate10K. Since the KITTI dataset includes outdoor scenes, the depth estimation module in the VRT is retrained on KITTI, which enables accurate depth estimation on large-scale outdoor scenes.

Comparison with Graph-RCNN

For the performance of VRT, we compare the predicted relationships with the ground-truth relationships from the offline model Graph-RCNN [318]. We adopt the hand-labeled images for the ablation study models the same as in the main paper.

Method	Input View	R@50	R@100
VRT	Original	31.2	33.5
KITTI [83]+VRD	Original	20.5	22.1
GRCNN [318]	Target	31.4	33.8

Table 4.11: Results evaluated with Recall@50 and Recall@100.

It is shown in Tab.4.11, when provided with the same original views as input, the VRT outperforms the VRD on the synthesized view. Furthermore, despite making predictions without the target view, our VRT achieves results comparable to Graph-RCNN [318].

Ablation Study

The ablation study results for models trained on KITTI are shown in Tab. 4.12. The performance of VRT on all relationships indicates the predicted relationships are accurately aligned with the ground truth. When only learned with the one-hot relationship label, the performance decreases a lot. Moreover, when relationship structure loss and relationship presentness loss are ablated, the performance also decreases.

Transformation sensitivity

	Recall@50	Recall@100
VRT	31.2	33.5
VRT w label	23.1	24.2
VRT w/o L_p	26.4	27.1
VRT w/o L_s	25.1	25.8

Table 4.12: Results and ablation study on KITTI.

Given the proposed approach is transformation-related, we conduct experiments here to test our VRT sensitivity with the transformation, which is trained on KITTI. Specifically, we split the testing frame pairs into three sets: T_1, T_2, and T_3, with angle $\leq 10°$ & translation ≤ 0.20, angle $\leq 15°$ & translation ≤ 0.25, and angle $\geq 15°$ *or* translation ≥ 0.25, respectively. The results on the hand labeling data are shown in Tab. 4.13. When tested on small-scale view transformation as in T_1, the performance is the best. T_2 and T_3 denote large-scale view transformation, so the accuracy decreases slightly. The model achieves good performance on all three views, which indicates the model is transformation insensitive.

	Recall@50	Recall@100
T_1	33.1	36.8
T_2	31.2	34.1
T_3	29.4	32.0

Table 4.13: Results of VRT with different transformations.

Transformation awareness

In this experiment, we conduct experiments on the hand labeling data to evaluate the performance of involving a transformation in the initial nodes feature, in edge representations, in attention calculation, and in the message-passing process. The model is trained on KITTI. As can be seen from Tab. 4.14, the message-passing one performs the best. This is because the transformation equivariance should be preserved by the node representations, edge representations, and attention, which can only be effectively captured by the message-passing process.

	Recall@50	Recall@100
T in nodes	29.8	31.3
T in edges	30.0	32.1
T in attention	30.4	32.8
T in message-passing	31.2	33.5

Table 4.14: Results of involving the transformation information in different stages of information aggregation.

4.4 Implementation details of downstream tasks

As mentioned before, we employ VRT across various tasks to assess its efficacy. Here, we are dedicated to illustrating how the acquired multi-view relationship representations from VRT can contribute to diverse tasks, such as visual relationship novel view synthesis, pedestrian intention estimation, and visual grounding.

4.4.1 Synsin [298]

As shown in Fig.4.8 the Synsin [298] firstly extract the visual features of the input image by the function f and then estimate the depth map by the function d. The two sets of features are then sent to projection together with the transformation matrix T to generate projected features. In Synsin+, as shown in Fig.4.9, each image is simultaneously sent to the VRT together with the transformation matrix T to predict the target relationship representations. The representations are then spatialized to the same size as the visual features in the Synsin by a relationship spatializer as in [314]. The spatialized representations are then concatenated with the visual features from the Synsin. The other parts are kept the same.

4.4.2 PixelNeRF [333]

The PixelNeRF [333], as shown in Fig. 4.10, firstly extracts visual features from the input view by a CNN encoder. Then, the coordinates together with the visual features are sent

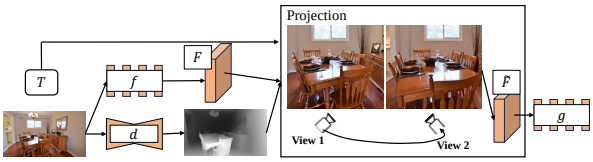


Figure 4.8: Framework of Synsin [298]

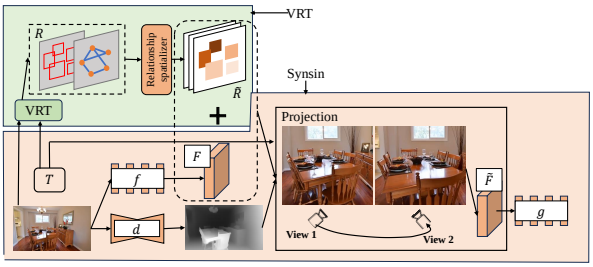


Figure 4.9: Framework of Synsin+.

to the model f to generate RGB and density values of the target view. In the PixelNeRF+, shown in Fig. 4.11, firstly predict the relationship representations of the target view by the original view input and transformation matrix T between the views. The relationship features enhance the rendering of the target view.

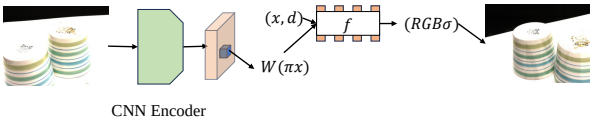


Figure 4.10: Framework of PixelNeRF [333]

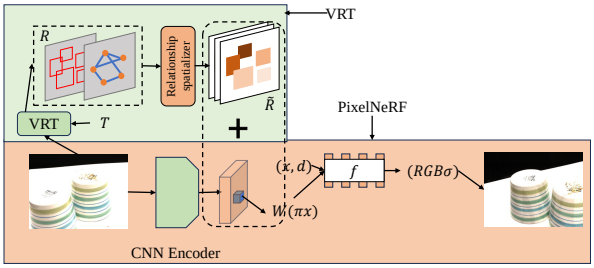


Figure 4.11: Framework of PixelNeRF+.

4.4.3 PIE [227]

The framework of the PIE [227] contains three parts: intention estimation, speed prediction, and trajectory prediction. When applying the VRT as a plug-in module to the PIE, only the intention estimation module is involved while the other modules remain the same. The modification is shown in Fig. 4.12, when PIE only takes the visual embeddings of each pedestrian-centered crop as inputs, we try to aggregate the relationship understandings from the pedestrian view into the intention estimation. Specifically, we first estimate the pedestrian gaze direction by the Gaze360 [122], the direction denotes the angle of the

pedestrian's gaze relative to the camera view. It can be converted to a transformation matrix T. With the T and input image, the target relationship representations are predicted. The representations are spatialized to the same size as the visual embeddings and then concatenated together.

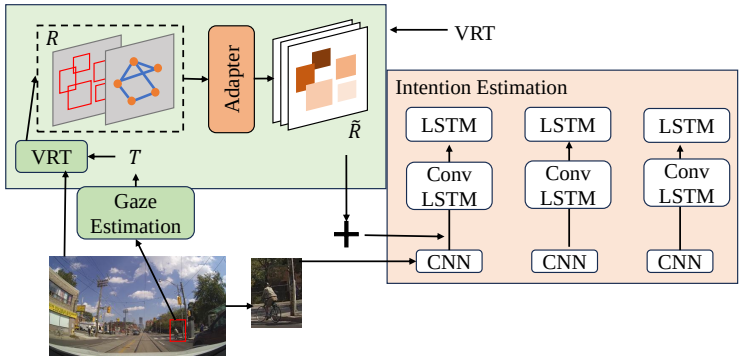


Figure 4.12: Framework of PIE+.

4.4.4 Visual grounding

As shown in Fig.4.13, the ReferIt3D [3] takes the 3d point cloud and text query as input and extracts visual and language features by a visual encoder and language encoder. Then the features are fused to locate the referred object. Since the ReferIt3D takes an unordered 3d point cloud as input, we have to designate an original view for each scene. Then for each query such as "Facing the window, the table left to the chair", we extract the view transformation from the original view to the view of "Facing the window". The ReferIt3D+, as shown in Fig. 4.14 firstly renders the original view, then sends it together with transformation matrix T to generate predicted relationship representations. The representations are then concatenated with the visual features.

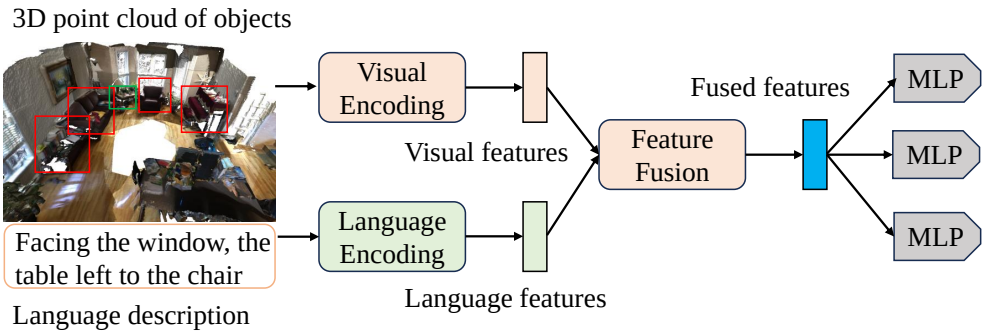


Figure 4.13: Framework of ReferIt3D [3]

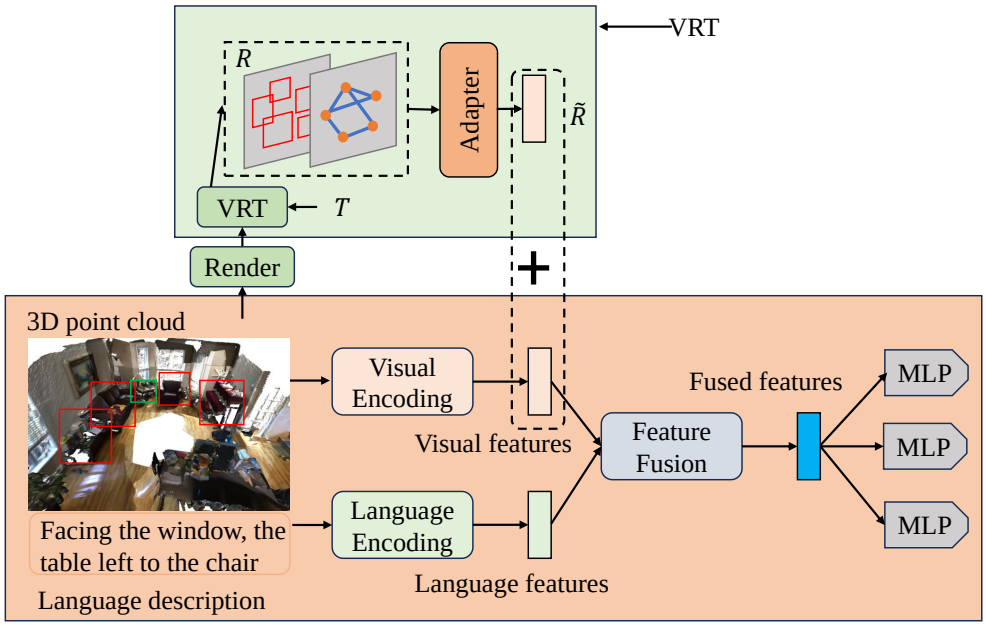


Figure 4.14: Framework of ReferIt3D+.

Chapter 5

Human-perceptual-preferences for novel view synthesis with sparse views

5.1 Introduction

The previous chapters introduce perceptual relationships to improve 3D understanding and quality. However, for single-object 3D models, perceptual relationships are not available. In this chapter, we instead strive to dig out human perception preference information to improve 3D rendering quality, especially for novel view synthesis.

Existing methods such NeRF [192] have achieved significant success for novel view synthesis, driven by the rapid advancement of neural rendering techniques. It highlights remarkable potentials in academic fields such as novel view synthesis, image editing, and digital human body[79, 342, 269], However, the performance of NeRF is significantly constrained by the availability of training views. In few-shot scenarios, NeRF is trained with limited supervised views, *i.e.,* 2 or 3, thus overfitting to the supervisions, leading to degraded quality in the generated novel views. The overfitting results in several artifacts: 1). The quality of particular novel views fluctuates during the training, undermining the

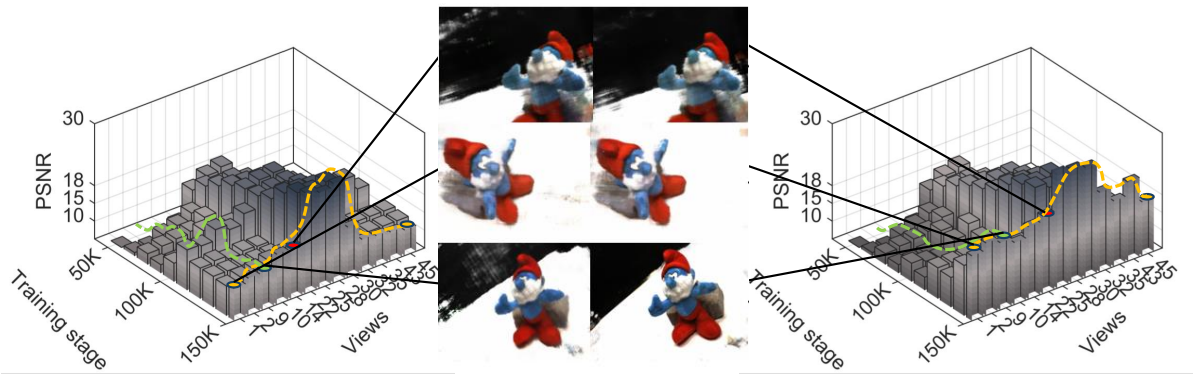


Figure 5.1: **Artifacts illustration**. Each bar plot presents the quality of renderings versus *training steps* and *viewpoints*. Our method addresses quality fluctuation (the green dot), distant view degradation (the yellow dot), and geometry inconsistency (the red dot). The left column of images is rendered by SPARF[282], and our method produces more perceptually satisfactory results as shown in the right column.

model's generalizability across multiple novel views. As shown in Fig. 5.1, the green dot and line exhibit the fluctuation along training steps. 2). Views farther from the supervision exhibit much inferior quality to the nearer ones, due to greater appearance variances. The distant view degradation results in an inconsistent cross-view experience, as seen with the yellow dots in Fig. 5.1, where the distant view suffers from underfitting. 3). Artifacts such as 'floaters' and 'background collapse', dubbed as geometry inconsistency, may arise in any novel view and persist throughout training as shown with the red dot view in Fig. 5.1.

Numerous methods have been proposed to alleviate the artifacts in few-shot NeRF. One line of research focuses on adding regularization to the photometric loss. However, these regularizations don't encourage perceptually decent renderings thus leading to photometrically correct but perceptually unsatisfactory viewing experiences. Prune et al. [282] impose photometric losses on homogeneously transformed views with learnable camera poses and estimated depth values. Their proposed multi-view objective and depth consistency loss won't generate perceptually satisfactory renderings once the estimated depth is incorrect. Jiawei et al. [319] proposed that frequency regularization improves the few-shot NeRF. Michael et al. [201] utilized the appearance regularization from a generative flow

model [50]. Again, these methods only aim to alleviate artifacts in the context of photometric loss rather than perceptual satisfaction. In contrast, human perception is more sensitive to different types of artifacts, involving human perception as regularization doesn't only guide alleviating the artifacts, but also prioritizes learning perceptual decent representations. The preceding work [144] exhibited that the meta-learned NeRF model with human perception as guidance implicitly improves perceptual satisfaction. We extend the work by explicitly introducing human perception into the objective function to guide the training dynamics. The intuition is illustrated in Fig. 5.2. The method [144] tilted the parameters of the vanilla NeRF model with meta-learning to learn more general parameters, which doesn't guarantee the rendered novel views are perceptually satisfactory. As a comparison, we explicitly formulate human perception into objective functions, which further tilted the parameters of the NeRF model to generate perceptually satisfactory renderings.

Human perception also helps learn the view transformation-robust (VT-robust) field representations. Brent et al. [326] exhibit that the VT-robust representations improve the quality of neural renderings and demonstrate that the VT-robust representations are not sensitive to noises of positions. Accordingly, we argue that introducing human perception as regularization encourages learning more VT-robust representations. The measurement of human perception grounds in that the distortion alters the statistical property of natural scenes, which is always represented as a mean opinion score (MOS) [198]. The photometric loss focuses on pixel-wise correction which over-penalizes the rapid color/luminance changes occurring in view transformation. In contrast, human perception focuses on overall scene naturalness rather than the detailed color/luminance changes. Moreover, the human vision system follows Weber's law, the magnitude of a just-noticeable luminance change ΔI is approximately proportional to the background luminance I for a wide range of [296]. Therefore, by adding human perception as regularization, the few-shot NeRF is less sensitive to abrupt color changes across views. We illustrate the concept in Fig. 5.3. For a trained NeRF, we first get the RGB values on coordinates with enough density and render them as point clouds. Then we attack the NeRF by adding noises to these coordinates, obtaining RGB deltas. The deltas reflect the robustness of the NeRF model to coordinates. Over-penalizing rapid color changes will cause larger deltas, denoting less VT-robustness. (a) of Fig. 5.3 exhibits the vanilla NeRF tends to be more erroneous than

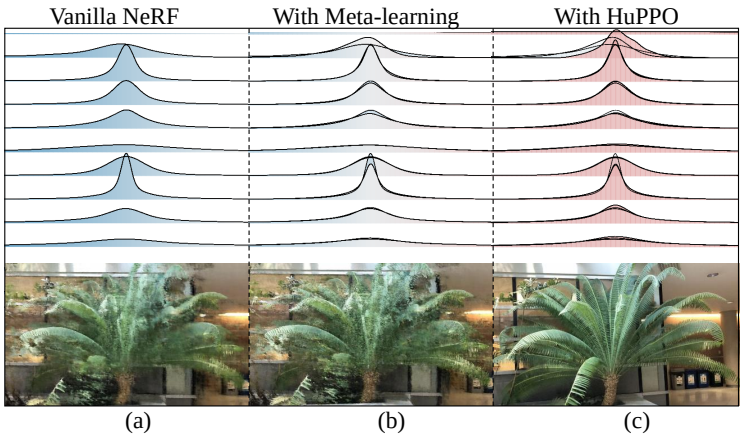


Figure 5.2: Human perception guides the parameters of the NeRF model (a) to progressively converge toward a perceptually satisfactory direction (c), which is better than a merely meta-learned model (b).

the ourmethod (b).

Nevertheless, It is challenging to construct a theoretically sound and practical framework to incorporate human perception into neural rendering. Because the methods for human perceptual quality measurement such as NR-IQA [359, 358, 357, 177] suffer from perceptual optimization distraction [355], which might generate adversarial samples, thus prohibiting directly adopting a pre-trained NR-IQA model as a loss function. Thanks to the development of human preference alignment methods [373, 308, 222, 38], the preference distribution can be learned without needing back-propagating a reward model. This merit enables employing the pre-trained NR-IQA method as a reward model to guide NeRF training.

In this chapter, we design a HuPPO framework to integrate image quality assessment (IQA) and visual preference optimization into few-shot neural rendering. The human-perception guidance helps learn perceptually satisfactory and VT-robust features to alleviate the mentioned distortions. Specifically, the proposed method contains three parts: 1) The candidate selection module. Since evaluating the perceptual preference on every view is infeasible, we instead count on the human perception to select reasonable candidates for enhancement; 2) HuPPO objective formulation. We formulate the human perception

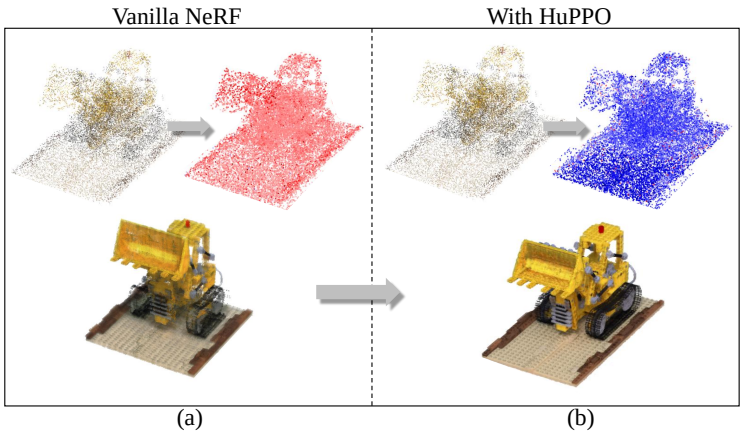


Figure 5.3: Human perception enhances the model's resilience against noises on view transformation. (b) shows that the human perception-guided field representations have fewer errors than vanilla NeRF (a).

into HuPPO objective given the selected candidates. 3) Meta-training pipeline. Since meta-learning helps extract shared data features, we add an extra H-step to learn the perceptually satisfactory features shared between original supervision and anchor viewpoints.

5.2 Background knowledge

5.2.1 Volume Rendering

A NeRF is a continuous function f, which is parameterized by a multi-layer perceptron (MLP) with θ, It maps a 3D position $x \in R^3$ and viewing direction $d \in S^2$ to volume density $\sigma \in R^+$ and RGB colors $c \in [0, 1]$ as $f_\theta(\gamma(x), \gamma(d)) = (\sigma, c)$, where γ is a predefined position encoding. Given the camera pose of the target perspective as P, the camera ray is defined as $r(t) = o + td$ along the direction d from the camera center o. The corresponding pixel values can be rendered with the known near and far bounds $[t_n, t_f]$.

By using alpha compositing, the NeRF render a 2D image as:

$$\begin{aligned} \hat{C}_\theta(r) &= \int_{t_n}^{t_f} T(t)\sigma_\theta(r(t))c_\theta(r(t),d)dt, \\ T(t) &= exp(-\int_{t_n}^{t} \sigma_\theta(s)ds). \end{aligned} \tag{5.1}$$

In rendering, discrete values are sampled along the ray between the near and far planes $[t_n, t_f]$ and approximated by numerical quadrature. Finally, the rendered 2D images are compared with the ground-truth by minimizing the mean square error:

$$L_{color} = \sum_{r \in R(P)} \left\| \hat{C}(r) - C_{gt}(r) \right\|_2^2, \tag{5.2}$$

where $R(P)$ is the collection of all camera rays. In the experiment, only a portion of rays are sampled for training.

5.2.2 RLHF & DPO

reinforcement learning with human feedback (RLHF) and direct preference optimization (DPO) are popular and effective methods to align language model output with human preferences. We first review the RLHF and DPO's pipelines as introduced in [373, 222].

RLHF includes 3 steps: supervised fine-tuning, reward learning, and reinforcement-learning optimization. For supervised fine-tuning, a pre-trained language model is adopted and fine-tuned with a dataset collected from specific tasks such as dialogue, and summarization. For reward learning, the preference data (x, y) is generated by human testers. Given any text prompt x, the language model generates an answer pair $\{(y_1, y_2)\}$, and the human's preference is annotated and represented as $(y_w \succ y_l|x)$, where y_w and y_l denote preferred and dispreferred answer, respectively. Then, the preference distribution is assumed to be modeled and learned by a Bradley-Terry reward model r^*:

$$p^*(y_1 \succ y_2|x) = \frac{\exp(r^*(x, y_1))}{\exp(r * (x, y_1)) + \exp(r^*(x, y_2))}. \tag{5.3}$$

For reinforcement-learning optimization, a new language model π_θ is fine-tuned from the original one π_{ref} under the guidance of the learned reward model with the objective:

$$\max_{\pi_\theta} \; -\mathbb{E}_{x,y\sim\pi_\theta(y|x)}[r_\phi(x,y)] - \beta \, \mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{ref}(y|x)]. \tag{5.4}$$

For **RLHF**, it's not necessary to learn a reward model, the preference distribution can instead be represented as:

$$p^*(y_1 \succ y_2|x) = \frac{1}{1+\exp(\beta \log \frac{\pi^*(y_2|x)}{\pi_{ref}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{ref}(y_1|x)})}, \tag{5.5}$$

where π^* denotes the language model fine-tuned from reference model π_{ref} on preference pairs $\{(y_2, y_1)\}$.

The objective of RLHF is formulated as:

$$\mathcal{L}_{DPO} = -\beta \, \mathbb{E}_{\mathcal{D}}[\log \sigma(\beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)})], \tag{5.6}$$

Where σ denotes a logistic function. To achieve RLHF, both the π_θ and π_{ref} need to evaluate the probability of the same preferred and dispreferred responses y_w and y_l. This is straightforward for a language model since its output format is the probability of the next token. By taking as input the first token of y_1 or y_2, the probability of $\pi(y_1|x)$ or $\pi(y_2|x)$ can be obtained once a full response is generated. However, it remains a challenge for vision tasks to model the probability of a response (RGB image). We will explain in the Sec 5.3.2. how to achieve visual preference modeling. Moreover, Rafailov et. al. [222] revealed that the weighted gradients might explain the mystery of RLHF:

$$\begin{aligned}
&\nabla_\theta(\mathcal{L}_{DPO}(\pi_\theta;\pi_{ref})) = -\beta \, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\alpha_{(\pi,x,y)} \cdot \mathcal{F}_{(\pi,x,y)}], \\
&\alpha_{(\pi,x,y)} = \sigma(\beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}), \\
&\mathcal{F}_{(\pi,x,y)} = \nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x).
\end{aligned} \tag{5.7}$$

The coefficient term α gets higher when the data are more dispreferred thus updating the parameters in more preferred directions. The gradient difference term $\mathcal{F}$ increases the likelihood of preferred data.

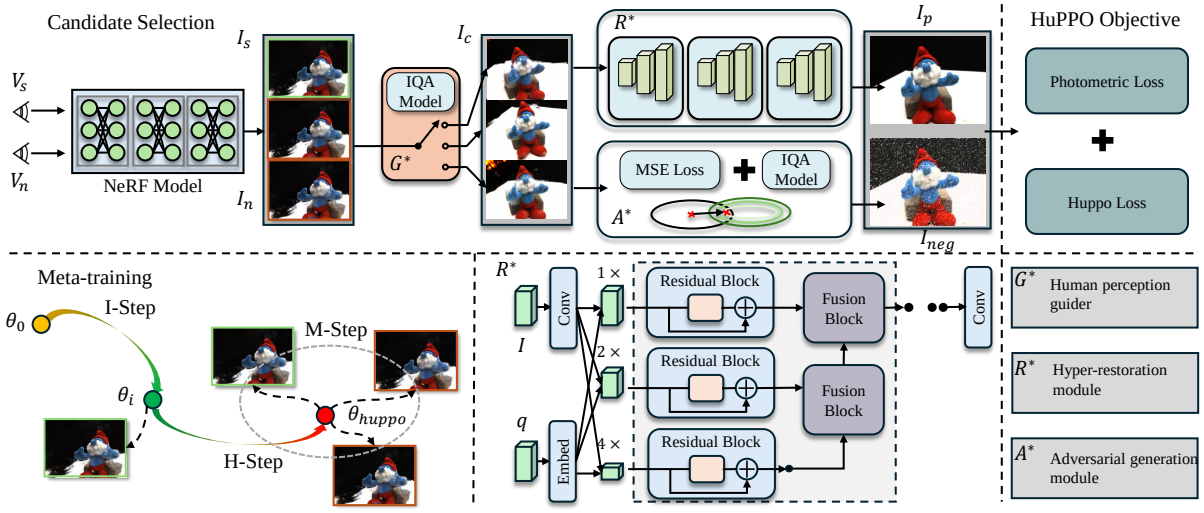


Figure 5.4: The proposed framework includes three modules: (1). candidate selection creates positive and negative renderings. The positive rendering is generated from a hyper-restoration module. The negative rendering is generated from perceptually adversarial attack; (2). HuPPO objective is constructed on the positive and negative renderings; (3). Meta-training first optimizes original supervisions in the I-step, then utilizes positive, negative, and supervised renderings to optimize the HuPPO objective in the H-step, and finally learns the general model in the M-step.

5.3 Method

Employing human perception as guidance for few-shot NeRF can get visually preferred renderings. To achieve this, we design a **Hu**man **P**erception **P**reference **O**ptimization (HuPPO) framework for Few-shot NeRF. In time order, the framework includes three parts: 1) Candidate Selection; 2) HuPPO Objective; and 3) Meta-training as shown in Fig. 5.4.

Before going into the details of the framework, we first explain the problem setup. When training a few-shot NeRF model f_θ, a set of supervision views $V_s = \{v_s : (x_s, d_s)\}$ is given, where x, d denote camera's positions and directions. The correspondingly rendered views are denoted as I_s. Our goal is to select novel views $V_n = \{v_n\}$ during the training

with guidance of NR-IQA model $Q_\phi(\cdot)$ and enhance the few-shot NeRF with HuPPO objective. Specifically, given the supervised views V_s, we first render novel views I_n at different training stages and then select candidate renderings $I_c \subseteq \{I_n\}$. By calculating the perceptual quality score, the human perception guider selects candidate renderings with quality fluctuation, distant view degradation, or geometry inconsistency. These candidate renderings are then processed with hyper-restoration and adversarial generation modules to generate positive and negative pairs I_p, I_{neg}. Secondly, we employ positive and negative pairs to construct HuPPO objectives. Finally, to guarantee the model prioritizes learning shared representations, we employ meta-training to search the optimal path that both fits the supervised renderings I_s and quickly generalizes to novel renderings I_n.

5.3.1 Candidate Selection

The *Candidate Selection* aims to obtain candidate positive and negative renderings to construct the HuPPO objective. It includes two steps, 1. candidate identification, and 2. positive and negative generation.

Candidate Identification. During the training, the novel views are rendered on the fly. A human perception guider helps identify which degraded views are legal as candidates. We adopt an offline NR-IQA model Q_ϕ as a human perception guider based on two considerations: 1. The NR-IQA algorithms are trained to align with human subjective perception, which better evaluates the naturalness of images; 2. The ground-truths of novel views are unavailable, so an NR-IQA model directly estimates the image quality without needing reference images.

Specifically, as shown in Fig. 5.4, we first render novel views $I_{nt} = f_\theta(V_{nt})$ under novel views $\{v_{nt} = (x_{nt}, d_{nt}) | v_{nt} \in V_{nt}, t \in (1, 2 \cdots T)\}$, where t denotes training stages. In practice, we get the novel renderings every 100 training steps. Then, by calculating the human-perception score $q_{nt} = Q_\phi(I_{nt})$, the views having fluctuated q_{nt}, severe degradation, and large distance are identified as candidate views. Note that the higher q_{nt} denotes the better quality, thus less degradation. Specifically, for human-perception score fluctuation, some renderings have large fluctuations where

$$|q_{n(t+\Delta t)} - q_{nt}| \geq \tau_f. \tag{5.8}$$

The Δt denotes the increment of training stages, and τ_f denotes the fluctuation threshold. For geometry inconsistency and distant view degradation, we consider both distance and degradation levels where

$$d(v_{nt}, v) \geq \tau_v, \quad q_{nt} \leq \tau_d. \tag{5.9}$$

The τ_v, τ_d denote view distance and degradation thresholds. Only when the degradation level is greater than the τ_d, and the distance is larger than the τ_v, the views are identified as candidates. With human perception as guidance, we select a set of candidate renderings $\{I_c\}$, which are then sent for positive and negative generation.

Positive Generation. The candidate renderings I_c are from raw NeRF that has different artifacts. We see the raw NeRF as an anchor model and hope to update it in a positive direction and away from the negative direction. So we synthesize positive renderings I_p based on the candidates via a hyper-restoration module. The hyper-restoration module can alleviate different levels of artifacts, thus implying a positive direction in the vicinity of the candidates.

Current deep learning-based image restoration networks only work on a fixed level of distortion, while the quality of NeRF renderings from different scenes and different training stages varies greatly. Therefore, we propose a hyper-restoration module that employs the human perceptual quality score as an indicator to guide the restoration network to select necessary features. As shown in R^* of Fig. 5.4, given a quality score q_c, and candidates I_c, the restored image is obtained by:

$$\begin{aligned} &I_p = R^*(I_c, q_c), \\ &R^*(I_c, q_c) = Conv(Fuse(Res(attn(Conv(I_c), Emb(q_c))))), \end{aligned} \tag{5.10}$$

where R^* denotes the hyper-restoration module. It firstly embeds renderings I_c and quality scores q_c. Then the embeddings are attended using an attention layer with $Emb(q_c)$ as queries and patched $Conv(I_c)$ as keys and values. It is noted that the $Conv(I_c)$ is downsampled to a half and a quarter of the size to capture global and local features. $Res(\cdot)$ denotes a residual block and $Fuse(\cdot)$ denotes a fusion block, idea borrowed from [346]. But instead of a long skip connection, we get output from the last $Conv$ layer. Because the input features should be selectively used according to the distortion level, while the long-skip connection blindly sends all information to the output. Using hypernetworks enables

effective image reconstruction, which requires only knowing the image quality score to achieve better image restoration at different degradation levels.

Negative Generation. When using the human perception, *i.e.* NR-IQA model, to guide optimization, it is necessary to avoid generating adversarial samples as explained in [356]. So we need negative samples to inform the NeRF away from the adversarial renderings. To this end, we search the adversarial directions in the neighbor areas of the candidates as done in [355]:

$$I^*_{neg} = \arg\max_{I_{neg}} -D(I_{neg}, I_c) + \lambda(Q_\phi(I_{neg}) - Q_\phi(I_c))^2. \tag{5.11}$$

I^*_{neg} denotes the searched adversarial rendering with a higher quality score but low perceptual quality. $D(\cdot,\cdot)$ denotes an Euclidean distance which constrains the search within the vicinity of the candidates I_c, here an MSE loss is used. Maximizing the $(Q_\phi(I_{neg})-Q_\phi(I_c))^2$ aims to find adversarial samples with high perceptual scores assessed by an offline NR-IQA model (Q_ϕ. λ balances the area of searching neighbors and adversarial neighbors.

5.3.2 Human Perceptual Preference Optimization

Given the positive renderings I_p, negative renderings I_{neg}, and candidates I_c, the HuPPO objective is constructed based on a direct preference optimization [222] formulation. The HuPPO aims to guide the model in approaching positive renderings and keeping away negative renderings. The positive renderings are more perceptually satisfactory while the negative renderings are adversarial samples. Based on the equation 5.5, we model the preference probability as:

$$p(I_p \succ I_{neg}|I_c) = \frac{1}{1+\exp\beta(\log\frac{\pi_\theta(I_p|I_c)}{\pi'(I_p|I_c)} - \log\frac{\pi_\theta(I_{neg}|I_c)}{\pi'(I_{neg}|I_c)})}. \tag{5.12}$$

By maximizing the preference probability $p(I_p \succ I_{neg}|I_c)$, the NeRF model's renderings gradually change from I_c to I_p. $\pi_\theta(y|x)$ denotes the probability of the learnable transfer model from x to y. $\pi'(y|x)$ denotes a reference transfer model. To understand how to connect the preference probability with human perception and the NeRF model, we have to clarify the details of the transfer model.

Transfer Model. As explained in section 5.2.2, the $\pi(y|x)$ denotes the probability of answer y given question x, which can be a large language model for natural language tasks. The preference probability can be obtained straightforwardly from equation 5.5, based on which the RLHF and RLHF work. For vision tasks, we interpret the $\pi(y|x)$ as the tendency of transferring from x to y. It's more probable to transfer when x and y are closer and y has good perceptual quality. We can model the transfer model in a maximum a posterior (MAP) estimation framework:

$$\pi(y|x) \propto \exp(-E(y|x)) = \exp(-(D(y,x) - \lambda q(y))), \tag{5.13}$$

where $E(y|x)$ denotes the Gibbs energy, $D(y,x)$ denotes the likelihood term, $q(y)$ denotes a prior term. As explained in [356], the likelihood term can be implemented with a mean squared error (MSE), and the prior term can be implemented with an NR-IQA model. The transfer model is consequently formulated as:

$$\begin{aligned} \pi_\theta(I_p|I_c) &= \exp(-D(I_p,I_c) + \lambda q(I_p)), \\ \pi_\theta(I_{neg}|I_c) &= \exp(-D(I_{neg},I_c) + \lambda q(I_{neg})), \\ I_c &= f_\theta(x_c, d_c). \end{aligned} \tag{5.14}$$

Here the candidates I_c are rendered by the NeRF model f_θ from camera positions and directions x_c, d_c, where θ denotes learnable parameters. The equation 5.14 manifests that the NeRF model tends to transfer to positive renderings with higher perceptual quality and move away from negative renderings with lower perceptual quality. Maximizing the preference probability in equation 5.12 allows us to get perceptually preferred NeRF model f_θ.

By taking the average logarithm value of the preference probability, we get the final **HuPPO objective**:

$$\begin{aligned} \mathcal{L}_{huppo} = -\mathbb{E}_{(I_p,I_{neg},I_c)\sim\mathcal{D}}(\log\sigma(\beta\log\frac{\pi_\theta(I_p|I_c)}{\pi_{ref}(I_p|I_c)} \\ -\beta\log\frac{\pi_\theta(I_{neg}|I_c)}{\pi_{ref}(I_{neg}|I_c)})), \end{aligned} \tag{5.15}$$

where σ denotes a logistic function.

5.3.3 Training scheme

Given the supervised renderings I_s, candidate renderings I_c, positive renderings I_p, negative renderings I_{neg} and HuPPO objective $\mathcal{L}_{huppo}$, the training scheme is developed with three steps: I-step, H-step, and M-step. As shown in Fig. 6.2, the I-step optimizes the NeRF f_θ on supervised renderings I_s with photometric loss as equation 5.2. The H-step optimizes the f_θ with HuPPO objective as equation 5.15. The M-step does a meta-learning update. For each iteration, we execute the I-step, H-step, and M-step once and repeat until the convergence.

I-step. In the I-step, the NeRF model is initialized with θ_0. Given the camera positions and directions x_s, d_s of supervised views, the optimal parameters θ_i is obtained with objective:

$$\theta_i = \arg\min_{\theta_0} \frac{1}{S} \sum_{x_s, d_s} \| f_{\theta_0}(x_s, d_s) - I_s^* \|_2^2 , \tag{5.16}$$

where S denotes the number of rays in a batch, the I_s^* denotes the ground truth color of view s.

H-step. In the H-step, we aim to learn the better human perception aligned parameters θ_h via minimizing $L_{huppo}(\theta_h)$, given the reference model f_{θ_i}, positive renderings I_p, negative renderings I_{neg}, and candidates I_c. The π' is initialized with the parameters θ_i from the I-step. So we have

$$\begin{aligned}
\pi_{\theta_h}(I_p|I_c) &= \exp(-D(I_p, f_{\theta_h}(v_c)) + q(I_p)), \\
\pi_{\theta_h}(I_{neg}|I_c) &= \exp(-D(I_{neg}, f_{\theta_h}(v_c)) + q(I_{neg})), \\
\pi'(I_p|I_c) &= \exp(-D(I_p, f_{\theta_i}(v_c)))), \\
\pi'(I_{neg}|I_c) &= \exp(-D(I_{neg}, f_{\theta_i}(v_c))).
\end{aligned} \tag{5.17}$$

Here we set λ to be 1, so it is omitted. $v_c = (x_c, d_c)$ denotes camera positions. Since the f_{θ_i} is optimized with photometric loss, the perceptual quality doesn't affect its transfer probability, so the right hand of π' doesn't include a prior term $q(\cdot)$. With the eq. 5.17,

the eq. 5.15 can be simplified as:

$$\begin{aligned}\mathcal{L}_{huppo}(\theta_h) = &- \mathbb{E}_{(I_p, I_{neg}, I_c)\sim\mathcal{D}}(\log \sigma[\\ &\beta(-D_{\theta_h} + q + D_{\theta_i}) \\ &-\beta(-\tilde{D}_{\theta_h} + \tilde{q} + \tilde{D}_{\theta_i})]),\end{aligned} \tag{5.18}$$

where $D_\theta = D(I_p, f_\theta(v_c))$, $q = q(I_p)$, $\tilde{D}$ denotes likelihood term of negative renderings.

To understand how the $\mathcal{L}_{huppo}$ achieves perceptual preferred optimization, we get the derivatives with $\beta = 1$:

$$\begin{aligned}&\nabla_{\theta_h}\mathcal{L}_{huppo}(\theta') = -\mathbb{E}[\\ &\sigma[(-D_{\theta_h} + q + D_{\theta_i}) - (-\tilde{D}_{\theta_h} + \tilde{q} + \tilde{D}_{\theta_i})]\cdot \\ &[\nabla_{\theta_h}(-D_{\theta_h}) - \nabla_{\theta_h}(-\tilde{D}_{\theta_h})].\end{aligned} \tag{5.19}$$

The derivatives exhibit the ideas of the HuPPO. Firstly, it encourages the model to approach the positive renderings and keep away the negative renderings when D_{θ_h} is small and $\tilde{D}_{\theta_h}$ is large. Secondly, the ratio $\sigma[(-D_{\theta_h} + q + D_{\theta_i}) - (-\tilde{D}_{\theta_h} + \tilde{q} + \tilde{D}_{\theta_i})]$ gets larger when the transfer model has wrong tendency. What is interesting is that the ratio term also avoids generating adversarial renderings. The adversarial vicinity is defined in [356] as images for which the fidelity terms $D_{\theta_h}, \tilde{D}_{\theta_h}$ are much smaller than a threshold, but the perceptual quality q, $\tilde{q}$ have a large discrepancy. In the equation 5.19, when $D_{\theta_h}, \tilde{D}_{\theta_h}$ are much smaller than q, $\tilde{q}$, the ratio becomes $\sigma(q - \tilde{q})$. So the optimization for renderings where $||\tilde{q} - q|| >> d$ will be punished, where d denotes discrepancy.

M-step. To quickly generalize to novel views, we can employ meta-training pipelines such as MAML[66] to learn a generic representation of NeRF. Different from the Meta-NeRF [274] which takes as task distribution of all scenes, our task distribution is measured on supervised plus positive and negative views. The insights are that we want to learn the shared features of different renderings of the same scene to avoid overfitting certain supervised views.

Like the MAML, we take the initial parameter θ_h from the H-step. Given the supervised renderings I_s, and positive renderings I_p, we hope the θ_h could be optimized to provide a

good starting point to the next I-step. Taking the θ_h from H-step as initial parameters θ, the objective is:

$$\theta = \arg\min_{\theta} \sum_{I} \mathcal{L}_I(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{I}}(f_\theta)}), \tag{5.20}$$

where I denotes the union of supervised and positive views. $L(\cdot)$ denotes photometric loss as euqation 5.16. α denotes the learning rate.

By iterating the I-step, H-step, and M-step, we finally get a perceptually preferred NeRF model. In the experiment, we can set different iterations for the I-step, H-step, and M-step to get optimal results.

5.4 Experiments

In this section, we first explain datasets and implementation details. Then, the proposed method is compared with state-of-the-art methods in Section 5.4.2 and we also explore how much these methods will be improved by being integrated into our framework. To deeply inspect the improvements over the explained distortions, we show a detailed analysis in Section 5.4.3. Then we also present a user study on the perceptual quality of renderings in Section 5.4.4. Finally, we conduct an ablation study to investigate the influence of using different NR-IQA methods, restoration modules, and the number of meta-views.

5.4.1 Datasets and Implementations

The proposed method is trained and evaluated on three datasets: the LLFF [249], Blender [192], and the DTU [113]. For LLFF, we use 10 scenes for testing phase optimization. In the training, we randomly select 3, 6, or 9 views as supervisions, and evaluate on other 8 viewpoints following the community standards [192]. For Blender, we take the same training and evaluating schemes as the LLFF on 6 scenes. For DTU, we select 10 scenes that contain clear objects for training and testing. Since we do not aim for generalizability as PixelNeRF [333], we fit each scene with an individual model. For other SOTA methods, we retrain them on the selected scenes, so the reported performances are somewhat different

from their papers. As for metrics, we report PSNR, SSIM[296], and LPIPS scores[354] on the evaluation viewpoints. Besides, since our training framework encourages more perceptually satisfactory rendering, we also report the perceptual quality score with the NR-IQA model [289]. Our proposed HuPPO can be inserted with other NeRF models, where only the training loss and meta-learning pipeline need to be added. So we compare with SOTA methods by implementing the HuPPO on their baselines. We train the hyper-restoration module out of the HuPPO on top of baseline [346]. To avoid data leakage from the same scene contents, we collected distorted images from 88 training scenes of DTU. For LLFF and Blender, we select one viewpoint for each scene. Note that these views are not accessible for the NeRF training and evaluation. We also take the degraded images from half-trained models to maximally enrich the restoration module training. When selecting candidate viewpoints we adopt the NR-IQA method LIQE [359]. In the NeRF training, for the DTU dataset, we uniformly downsample all images to 400×300 and randomly sample 1024 rays in batch size to calculate the loss. For the LLFF dataset, we adjust the resolution to 1/8 of the original size and randomly sample 2048 rays at each optimization iteration. All experiments are trained on an A100.

5.4.2 Comparisons

Here, we report the experimental results in both quantitative and qualitative ways. Since our training framework can embed with any NeRF model, we compare with NeRF [192], Pixel-NeRF [333], Dpeth-NeRF [49], Reg-NeRF [201], and SPARF [282], by replacing the NeRF backbone with these methods.

LLFF dataset. The table 5.2 presents the results of the SOTA methods and the enhancement with our HuPPO framework on the LLFF dataset [249]. The results show that in both 3, 6, and 9 views evaluation, the introduction of HuPPO leads to good improvements on both PSNR, SSIM, and LPIPS. To demonstrate that HuPPO helps improve perceptual quality, we also compare the perceptual quality scores evaluated by the NR-IQA model LIQE [359]. The last 3 columns present the NR-IQA scores, scaling to $(0, 1)$, we find that HuPPO improves more on the pixel-NeRF, we analyze that is because they introduce semantics information into the NeRF model, and the LIQE also adopts the pre-trained

Method	Type	PSNR ↑			SSIM↑			LPIPS ↓			LIQE ↑		
		3	6	9	3	6	9	3	6	9	3	6	9
NeRF	-	14.41	17.51	21.45	0.52	0.63	0.75	0.31	0.25	0.14	0.45	0.52	0.61
NeRF(h)		16.20	19.74	22.37	0.54	0.64	0.80	0.29	0.25	0.13	0.47	0.53	0.60
Pixel-NeRF	Semantic	16.82	19.11	20.40	0.70	0.75	0.76	0.27	0.23	0.22	0.54	0.56	0.57
Pixel-NeRF(h)		17.54	19.60	21.13	0.72	0.78	0.80	0.24	0.21	0.19	0.56	0.59	0.59
Depth-NeRF	Depth	16.50	20.50	22.20	0.54	0.73	0.77	0.48	0.31	0.26	0.50	0.58	0.62
Depth-NeRF(h)		17.21	21.84	22.97	0.61	0.79	0.82	0.45	0.30	0.26	0.54	0.60	0.64
Reg-NeRF	Regularization	20.11	22.34	24.21	0.65	0.74	0.80	0.35	0.27	0.22	0.58	0.62	0.70
Reg-NeRF(h)		20.34	23.73	25.13	0.78	0.84	0.88	0.37	0.25	0.21	0.57	0.63	0.72
SPARF	Depth+	18.30	23.24	25.75	0.78	0.87	0.91	0.21	0.12	0.08	0.56	0.64	0.75
SPARF(h)	Regularization	19.40	23.58	25.82	0.82	0.90	0.94	0.16	0.12	0.10	0.56	0.62	0.76

Table 5.1: Comparison with SOTA methods on DTU. We retrain and evaluate the SOTA methods on 3, 6, and 9 viewpoints and report the performances of PSNR, SSIM, LPIPS, and NR-IQA. The "Type" column exhibits the features or regularizations of the methods. The light gray rows present methods enhanced with the HuPPO framework, e.g. Nerf(h). the ↑ denotes the higher the better and the ↓ denotes the lower the better.

Method	Type	PSNR ↑			SSIM↑			LPIPS ↓			LIQE ↑		
		3	6	9	3	6	9	3	6	9	3	6	9
NeRF	-	13.61	16.70	18.45	0.28	0.43	0.51	0.56	0.40	0.31	0.46	0.50	0.52
NeRF(h)		16.82	18.27	20.33	0.44	0.53	0.66	0.47	0.35	0.26	0.51	0.54	0.60
Depth-NeRF	Depth	19.80	22.79	24.58	0.68	0.69	0.70	0.41	0.39	0.38	0.56	0.61	0.70
Depth-NeRF(h)		19.91	22.80	24.58	0.70	0.72	0.72	0.40	0.39	0.37	0.56	0.62	0.71
Reg-NeRF	Regularization	19.63	23.73	25.13	0.61	0.78	0.82	0.31	0.19	0.16	0.55	0.65	0.73
Reg-NeRF(h)		19.72	23.80	25.44	0.68	0.80	0.84	0.30	0.18	0.16	0.56	0.65	0.73
SPARF	Depth+	20.20	23.35	24.40	0.63	0.76	0.80	0.24	0.20	0.18	0.60	0.65	0.68
SPARF(h)	Regularization	20.41	23.51	24.69	0.68	0.80	0.83	0.21	0.18	0.17	0.62	0.65	0.69

Table 5.2: Comparison with SOTA methods on LLFF.

Method	Type	PSNR ↑			SSIM↑			LPIPS ↓			LIQE ↑		
		3	6	9	3	6	9	3	6	9	3	6	9
NeRF	-	12.53	15.21	16.48	0.43	0.51	0.56	0.40	0.31	0.30	0.39	0.45	0.49
NeRF(h)		13.87	16.41	17.22	0.51	0.57	0.59	0.35	0.29	0.30	0.41	0.47	0.49
Pixel-NeRF	Semantic	15.47	18.20	19.44	0.68	0.72	0.74	0.25	0.21	0.21	0.47	0.52	0.56
Pixel-NeRF(h)		15.60	18.33	19.49	0.70	0.73	0.74	0.22	0.19	0.18	0.50	0.54	0.57
Depth-NeRF	Depth	15.70	20.11	21.34	0.50	0.68	0.74	0.40	0.33	0.28	0.52	0.58	0.61
Depth-NeRF(h)		15.94	20.37	21.50	0.54	0.70	0.74	0.38	0.32	0.29	0.54	0.58	0.62
SPARF	Depth+	19.57	22.41	23.40	0.58	0.73	0.78	0.22	0.21	0.18	0.57	0.61	0.68
SPARF(h)	Regularization	19.80	22.64	23.58	0.66	0.76	0.80	0.20	0.18	0.16	0.58	0.62	0.70

Table 5.3: Comparison with SOTA methods on Blender.

backbone of the visual-language model. In another aspect, the Pixel-NeRF doesn't perform well on the llff dataset because it is pre-trained on the DTU dataset, which contains more about single object contents. So the distribution of data in DTU is much different from the llff. After being enhanced with the HuPPO framework, which provides perceptual guidance on more general data, the Pixel-NeRF progresses remarkably, demonstrating the effectiveness and generalizability of the HuPPO.

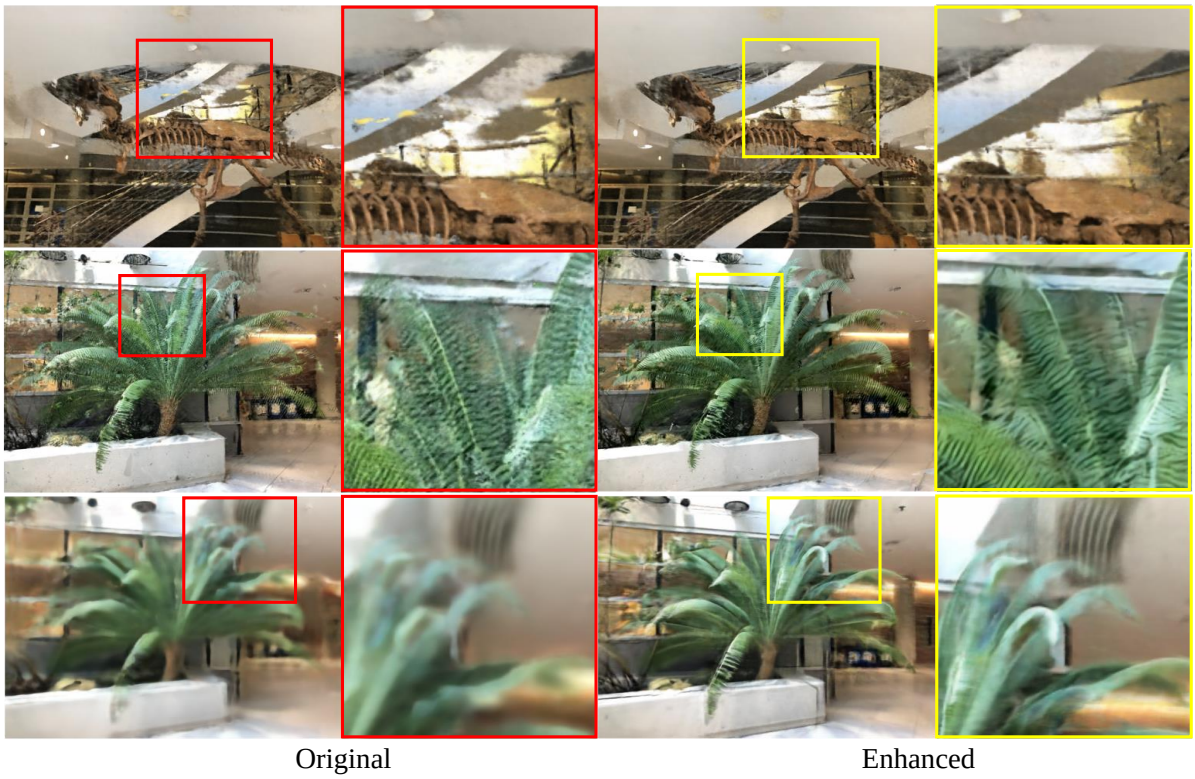


Figure 5.5: Qualitative comparisons of SOTA methods NeRF [333] (first row), Depth-NeRF [49] (second row), and Sparf [282] (third row). The red-boxed figures present the original models' results while the yellow-boxed figures present the enhanced results with HuPPO optimization.

DTU & Blender datasets

contain simple objects while the LLFF contains natural scenes. Quantitative results of the DTU and Blender datasets are summarized in Table 5.1 and Table 5.3, where our

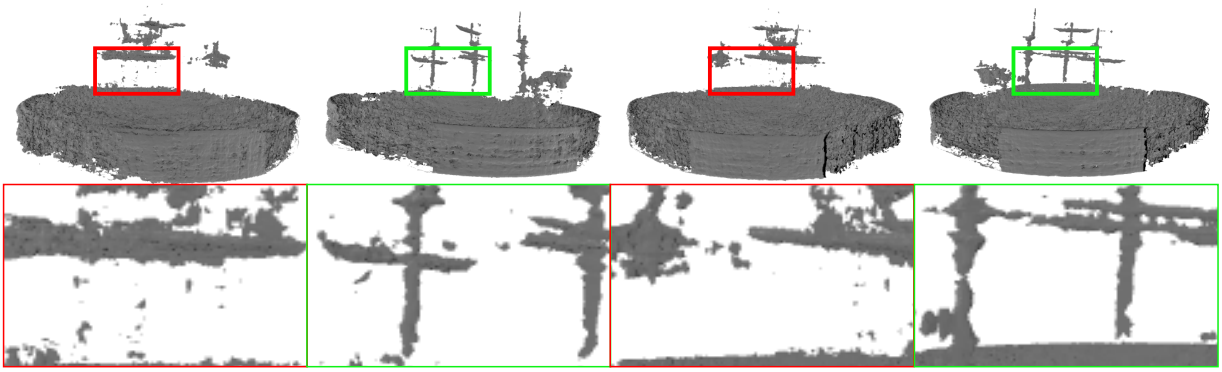


Figure 5.6: Geometry rendered by marching cubes [172]. The first and the third columns plot two different views of the mesh learned by NeRF [192] on 3 supervision views. The second and fourth columns present two views of the mesh learned by the NeRF enhanced with the HuPPO.

method achieved the best overall performance. For the DTU, we don't compare masked results as in [192, 201], because the aforementioned artifacts are possible to occur in the background. Compared to frequency and geometric regularization methods[201], our results show a 0.2−1.4 improvement in PSNR even when there are only three training perspectives. Compared to the method of adding additional supervisory information[282], our method still outperforms them in terms of performance and achieves an average increase of 0.5 in PSNR. In addition, compared to Pixnerf[333], our method renders images with less damage and leads by an average of over 0.6 on PSNR, further proving that our method can introduce more scene information.

Qualitative results are shown in Fig. 5.5. We compare the visual results of NeRF, Depth-NeRF, and Sparf trained with 3 views. Each method is enhanced with the HuPPO framework. For each method, the red-boxed one shows the results of the SOTA methods, and the yellow one exhibits the HuPPO results. Observing the NeRF-NeRF(h) trained on 3 views, the HuPPO successfully removes the 'floaters', i.e. geometry inconsistency. For Depth-NeRF on 3V, the HuPPO generates more perceptually satisfactory renderings of the 'fern'. Besides, the HuPPO also avoids overly noisy or blurry rendering as shown in Sparf.

5.4.3 Distortions removal

To understand how the HuPPO helps remove the mentioned distortions: "quality fluctuation", "distance view degradation", and "geometry inconsistency", we investigate the structural quality, appearance quality, and training dynamics of the few-shot NeRF model and the one enhanced by the HuPPO.

Geometry Inconsistency. To analyze how the HuPPO alleviates the geometry inconsistency, we adopt the marching cubes [172] algorithm to construct the meshed surface of the learned fields. The Fig. 5.6 compares the learned meshes by NeRF [192] and the HuPPO. It is observed that with the HuPPO as regularization, the learned geometry is more complete and has fewer floaters. This is because the HuPPO encourages the optimization of the NeRF in a perceptual, satisfactory direction. The unsatisfactory direction such as incomplete structure is discouraged by penalizing the adversarial sample generated by equation 5.11. So the floaters and incomplete structure which will be recognized as low perceptual quality are alleviated.

Distant View Degradation. We further inspect how the HuPPO helps improve distant view degradation. As shown in Fig. 5.7, we render the neighboring views of the supervised one and compare their visual quality. We select two distant novel views, the red cameras in the left picture, and one supervised view, the green camera. The enhanced distant views have more clean details. Further, we also check the perceptual qualities of the distant views as shown in the left picture of Fig 5.8. It is observed that for Sparf [282] and DsNeRF [49], the perceptual quality suffers the distant view degradation, where the supervised view (#5) has a much higher score than its neighbors. After being enhanced with HuPPO (Sparf(h)), the quality difference between the supervised view and the neighboring views is less, which provides a smoother viewing experience.

Quality fluctuation. To visualize how the HuPPO decreases quality fluctuation, we analyze the training dynamics of the few-shot NeRF [192] and the HuPPO. As shown in Fig. 5.9, we select the two nearest neighboring views of the supervised one and observe their quality fluctuation during the training. Note that these two views have reference counterparts so we can calculate their PSNRs. The reason for selecting two nearest neighbors is that they experience the highest level of quality fluctuation. The training stages

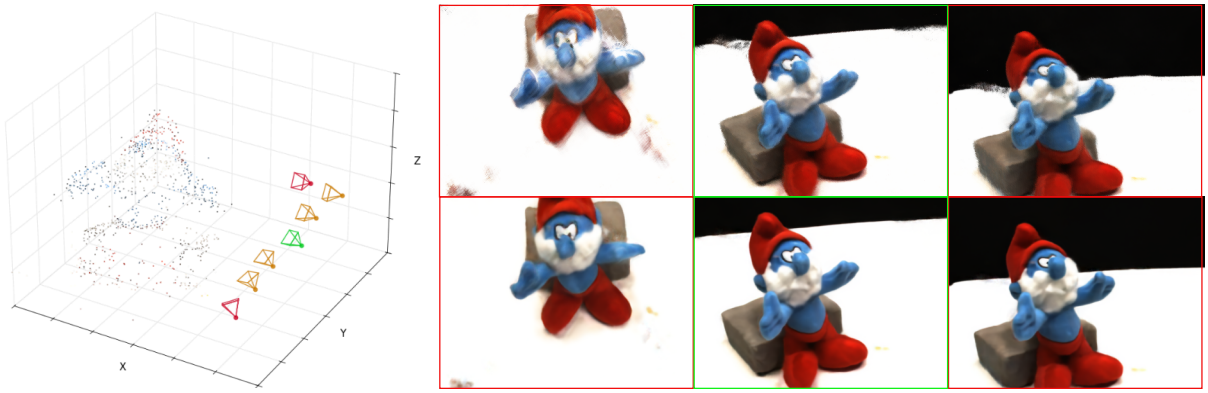


Figure 5.7: Distant view degradation. The left plot illustrates which views we select. The green view camera represents the supervision view, and the brown and red cameras denote novel views. The top 3 renderings are from the sparf [282] trained on 3 supervision views, the bottom 3 are enhanced with HuPPO.

from 1 to 10 represent 10 milestones in the training, each corresponding to 500 iterations. It is observed that the Few-shot NeRF suffers from quality fluctuation starting from about 30000 iterations, since when the quality drops down. The HuPPO alleviates the quality fluctuation and gets better quality. Even though the HuPPO doesn't perform better at the early training stages, as shown in the "our1" curve, it finally surpasses the quality fluctuated one, as shown in the "NeRF1" curve. This demonstrates the robustness of the HuPPO and the damage of quality fluctuation.

5.4.4 User study

As stated in Sec. 5.1, the HuPPO contributes to learning perceptual decent features. However, justifying it by only providing NR-IQA scores is not enough because the NR-IQA model is involved as regularization in the training. So we do a user study to help identify which rendering of NeRF or HuPPO has higher quality and rank them in three levels (good, neutral, bad). In the user study, we randomly render 5*10 views from 5 NeRF models and ones enhanced with HuPPO. Then we invite 15 subjective testers to score the quality of the renderings. The three levels of scores are mapped to $(1, 2, 3)$ and then averaged over

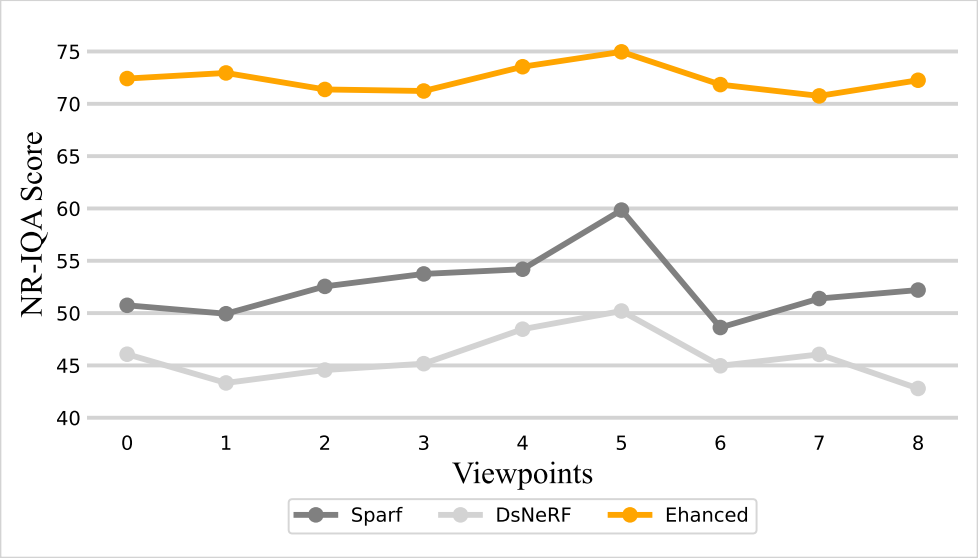


Figure 5.8: Distant view degradation. The perceptual quality of distant views of Enhanced Sparf is better than that of Sparf and DsNeRF.

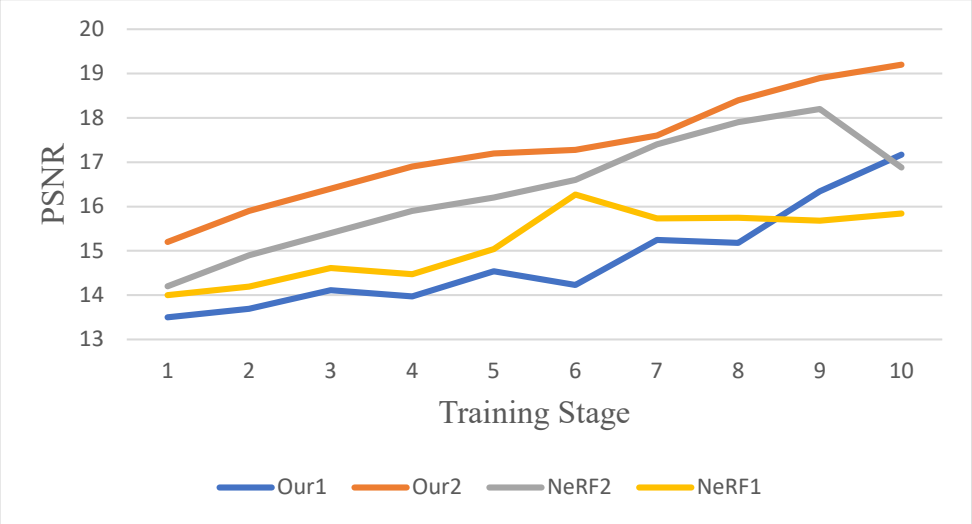


Figure 5.9: Quality fluctuation: the PSNR v.s. training stages of the two nearest neighbors to the supervised views are denoted as "NeRF1" and "NeRF2". The dynamics of HuPPO are denoted as "Our1" and "Our2".

	Scenes				
	scan82	fern	trex	flower	hotdog
NeRF	2.13	2.33	2.47	2.40	2.06
HuPPO	2.40	2.60	2.67	2.67	2.27

Table 5.4: Subjective test for perceptual quality assessment of 5 scenes.

testers. We report the average scores in Table. 5.4 on 5 scenes. The HuPPO gets better perceptual quality than the few-shot NeRF model, which demonstrates that the HuPPO encourages the NeRF model to learn perceptually satisfactory features.

5.4.5 View-transformation robustness

The HuPPO also contributes to learning the view-transformation robust features. Here the features refer to the intermediate outputs of the NeRF model, whose size might be $N \times L$. The N denotes the number of valid points, of which the σ (in equation 5.1) is larger than a threshold, and L denotes the length. However, it is not straightforward to visualize the VT robustness of the learned features since it is a high-dimensional vector. We instead measure the perceptual quality robustness of different NeRF models given certain levels of view perturbation. The idea is if the NeRF model exhibits less quality variation given different view perturbations, then it proves the model learns VT robust features. As shown in Table. 5.5, we impose 5 different levels of random view perturbation onto the NeRF [192] and HuPPO and report the average and standard deviation of the NR-IQA scores. For each perturbation level, the scores are averaged over 10 randomly selected perturbations on 3 NR-IQA models [358, 194, 359], and their outputs are mapped to $0 \sim 100$. It's observed that 1. the HuPPO exhibits less variation towards view perturbation; 2. the larger perturbation influences not only the variation but also the average quality, where HuPPO experiences less decrease than the NeRF. The results demonstrate that HuPPO learns more robust features.

	View perturbation				
	$0-5°$	$5-10°$	$10-15°$	$15-20\circ$	$20-25°$
NeRF	46.3 (4.7)	45.5 (5.4)	44.7 (6.3)	42.9 (9.5)	42.7 (10.3)
HuPPO	57.8 (3.5)	57.1(4.1)	56.4 (4.9)	55.1 (5.5)	54.5 (6.2)

Table 5.5: Ablation study of results under different view perturbation.

	Training Views				
	3	3+1	3+2	3+3	3+4
PSNR	17.40	18.36	18.40	18.47	18.58
SSIM	0.79	0.80	0.81	0.82	0.84
LPIPS	0.20	0.23	0.22	0.21	0.20

Table 5.6: Ablation study of results for different candidate views.

5.4.6 Ablation study

In Table. 5.6, We randomly select a scene from the DTU dataset and ablate the number of increased training views. Here, we have selected models trained from three different views with corresponding ground truth as the baseline, while the other increased views are images obtained through our method with certain damage. It can be seen that within a certain range, the performance of the NeRF model is gradually improving with the addition of new views. However, when there are too many damaged image views, the performance of the model will become inferior to the baseline. This may be because a large amount of damaged images have caused too much noise to interfere with the accuracy of the model. Therefore, in the experiment, taking into account the quality of the rendered image and the convergence speed of the model, we adopted the scheme of adding one view. In Fig. 5.8, we compare different NR-IQA models (LIQE [359], BRISQUE [194], NIQE [195], HyperIQA [268], Golestaneh [91]) by mapping them to $0 \sim 100$. The results show that the LIQE has the most similar tendency as the PSNR, so it is a reliable method for candidate selection.

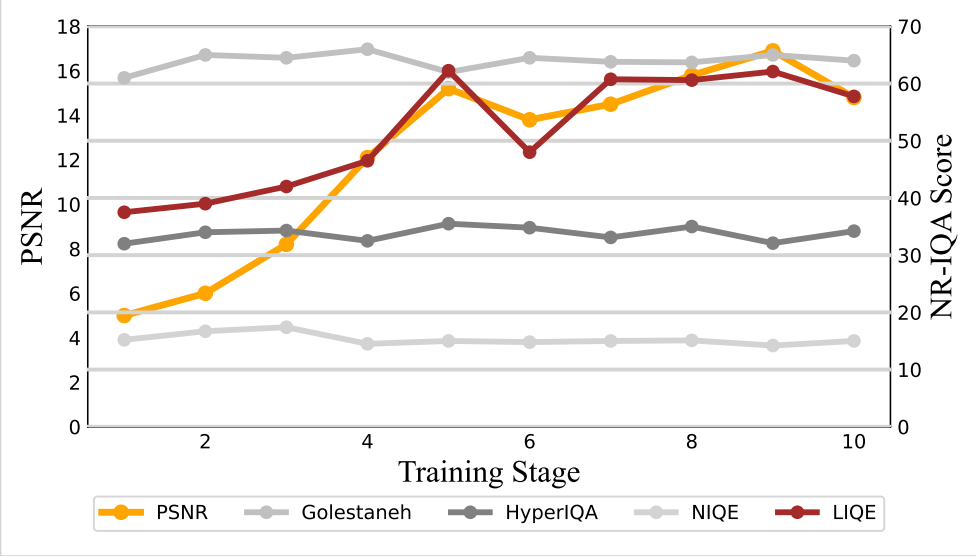


Figure 5.10: Ablation study on the performance of different NR-IQA models.

Chapter 6

Efficient and General 3D Representations for 3D Data Generation

The above chapters introduce the effectiveness of extracting perceptual representations to improve 3D understanding. In another aspect, the efficiency and generalizability of representation are also major concerns in making 3D models applicable. In this chapter, we introduce efficient and general 3D representations and apply them to various 3D data, including point clouds, and neural radiance fields. We also evaluate the ideas on 2D images. Specifically, we focus on representing 3D data with INR.

6.1 Introduction

INRs are powerful tools to represent complex data as continuous functions with neural network parameters. This functional representation is agnostic to the underlying data modality, making INR a universal way to represent diverse data types, including audio, images, and 3D volumes. Unlike conventional discrete data structures with explicit formats, INRs implicitly encode data in the parameter space. This allows for more detailed,

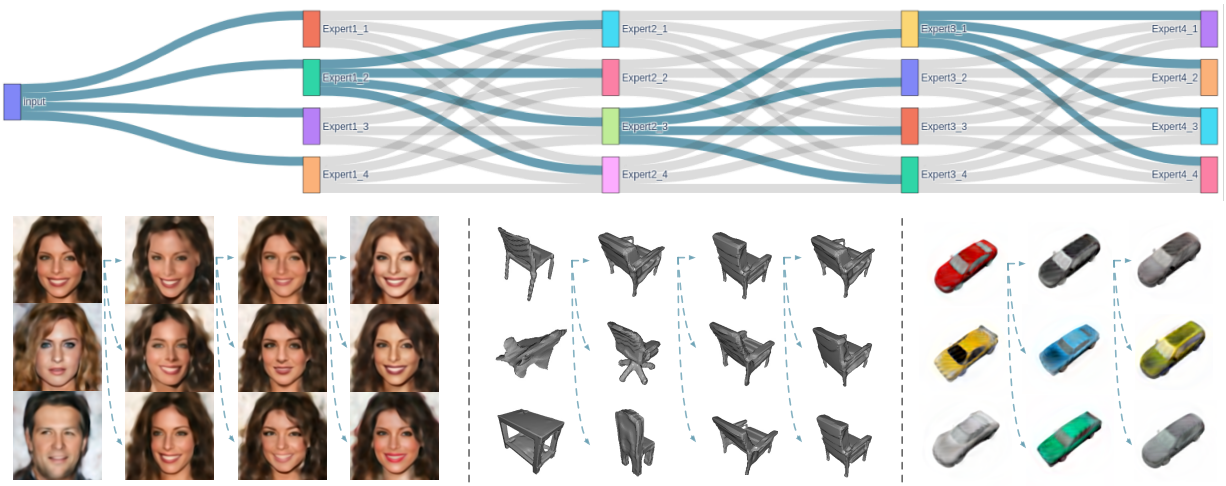


Figure 6.1: Hierarchical generation of universal data modality. *Top*: expanded representational capabilities of a Layers-of-Experts INR model. *Bottom*: aligning these capabilities with hierarchical data patterns enables precise control over the generation process. Each column presents generated samples resulting from a divergence in routing at a specific layer, with arrows indicating the shared routing in the preceding layers.

compact, and flexible representations. Recent advancements in the quality and efficiency of INRs greatly enhanced their performance across various applications, such as image fitting [257, 56], video compression [25, 24, 152], shape modeling [365, 191], novel view synthesis [193], and beyond.

The generalization ability of INRs, however, remains limited, since INRs are typically trained to overfit individual data instances. Keeping in mind that all data information is embedded in INR's parameters, the parameter space acts as a hidden manifold that captures the structures and variations of data. Building on this concept, a recent line of work [209, 20, 57, 332] has focused on modeling the parameter distribution to generalize INRs. A common approach is to adopt a two-stage framework. First, a collection of INRs is fitted to each data instance. Given a collection of data $\{(\mathbf{x}, \mathbf{f})\}$, each INR $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{F}$ is learned by finding weight $\boldsymbol{\theta}$ to minimize the distance between $f_{\boldsymbol{\theta}}(\mathbf{x})$ and $\mathbf{f} \in \mathcal{F}$. $\mathbf{x} \in \mathcal{X}$ denotes coordinates (*e.g.* pixel locations) and $\mathbf{f} \in \mathcal{F}$ denotes signals (*e.g.* RGB values). Second, the weight distribution $p(\boldsymbol{\theta})$ is learned through generative models. Since the raw

INR parameters are intractably high-dimensional, a condensed latent vector $\mathbf{h}$ is often used to represent each INR via modulation [57]. This simplifies the task of learning $p(\boldsymbol{\theta})$ by transforming it into the surrogate task of learning $p(\mathbf{h})$, which exists in a more manageable lower-dimensional space.

Despite the various modeling approaches, these methods learn the joint distribution of all parameters of an INR, treating each INR's parameters indiscriminately via a flattened latent. This approach overlooks the inherent hierarchical structure of INRs, particularly in multilayer perceptron (MLP)-based architectures such as SIREN [257], where each layer's representation capacity depends conditionally on the previous layers. As a result, these methods fail to capture the hierarchical patterns in the data. Taking the facial image as an example, hierarchical semantics refer to progressively detailed facial characteristics, such as overall facial shape, expression, and shape of eyes. Meanwhile, the parameters of INRs exhibit layer-wise expanded representation capacity [343], determined by its hierarchical structure. However, learning the joint distribution with flattened parameters ignores the correspondences between the INRs' expanded representation capacity across layers and the hierarchical patterns in the data. This misalignment brings two challenges: (1) The generalizability of INRs to unseen data is impaired. The joint distribution learning encodes the entangled semantics together into one flat latent, where the co-occurrence of certain semantics is inevitable. Therefore, the diversity of the generated data is limited. (2) Control over the generated content is limited. While generative models, such as latent diffusion, are employed with INRs to generate new data, they cannot link the sampled noise to the expected semantics in the output.

To address these challenges, we propose a novel approach that models the hierarchical structure of INRs, promoting layer-wise control in the generation process. Our method starts by training a collection of INRs on a dataset. Each layer of an INR is parameterized as a mixture of experts (MOE) layer to increase expressivity, where a set of expert weights and a latent vector are learned. The experts at each layer are shared across the dataset, while the latents are data-specific, routing the data flow and modulating the contribution of experts. Layers of MOE are cascaded to form an INR, which we call a LOE network. Consequently, a LOE with L layers will have L latents adapted to the fitted data, effectively capturing and relating its complex patterns with layers of latents. By modeling the

conditional dependencies of these layer-wise latents with our proposed HCDM, we maintain the hierarchical structure of INRs. This unlocks a controllable generation process, aligning the layers in INRs with the hierarchies of data semantics for the first time. As illustrated in Fig. 6.1, the data flow for a generation process resembles a tree-like structure: the routing in the next layer is constrained by the paths in previous layers, allowing full control of where a different routing strategy should be explored. Early deviations in routing lead to significant semantic differences in the generated content, while a later deviation results in minor differences in details.

6.2 Background knowledge

In this section, we introduce INRs and their generalization ability while highlighting their connections to our proposed approach. We also analyze the inherent hierarchy in INR's parameters.

6.2.1 Implicit Neural Representation and Generative INRs

INRs parameterize data such as audio, images, video, and 3D voxels as mappings from coordinates to signals, enabling a unified framework for various data modalities [85, 306]. Remarkable progress has been made to enhance the representation quality, efficiency and compactness for audio [377, 176, 267, 137], images [257, 64, 32, 312, 240], 3D contents [193, 13, 279, 205, 365, 237], and videos [25, 152, 315]. Despite performing well on different modalities, INRs struggle to generalize to multiple and unseen data, as each instance is typically overfitted with a separate MLP. To address this, two key strategies have emerged: (1) learning content-specific input features [333, 107, 138] and (2) modulating network parameters with hyper-latents [187, 290, 58, 124]. Generative models [92, 104] further extend INR's capability to generate new data. GRAF [248] and GIRAFFE [202] generate shape and appearance codes from noise, which are combined with coordinates to construct scenes. [63] use a diffusion model to generate INR weights. [59, 54, 57, 15] learn the distribution of latents that map to or modulate INR parameters, and generate new content

by sampling in the latent space. mNIF [332] further enhances the expressivity of INR via model averaging. These methods essentially model the distribution of INR parameters $p(\boldsymbol{\theta})$ by learning latent distributions $p(\boldsymbol{h})$, but fail to capture the layer-wise hierarchical structure of INR parameters, limiting their ability to accurately model distributions and control generation, which will be discussed in Sec. 6.2.2. Building on the latent modulation approach, we introduce a hierarchical controllable diffusion model, capturing dependencies between layer-wise latents for improved generalization and control.

6.2.2 Hierarchy Analysis of INR

In this section, we review the INR architecture and analyze its inherent hierarchical representation ability. Using the SIREN [257] as an example, a two-layer SIREN is generally formulated as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_2 \sin(\mathbf{W}_1 \cdot \gamma(\mathbf{x})), \boldsymbol{\theta} = [\mathbf{W}_1, \mathbf{W}_2], \tag{6.1}$$

where $\gamma(\mathbf{x}) = \sin(\Omega \cdot \mathbf{x}), \Omega \in \mathbb{R}^{c_1 \times c_{in}}$ denotes positional embedding of coordinates $\mathbf{x}$, $\mathbf{W}_2 \in \mathbb{R}^{c_{out} \times c_2}$, $\mathbf{W}_1 \in \mathbb{R}^{c_2 \times c_1}$ denote the parameters of each layer. The bias is omitted for simplification. From the perspective of Fourier-Transformation, the input frequency domain Ω is composed of c_1 frequency basis, $\Omega = [\Omega_0, \Omega_2, \cdots, \Omega_{c_1-1}]$. According to the [275] and [343], an MLP layer with periodic activation $\sin(\cdot)$ only expands the input frequency basis in a sparse and limited bandwidth. The equation 6.1 can be reformulated as:

$$\begin{aligned} f_{\boldsymbol{\theta}}(\mathbf{x}) &= \sum_{w' \in \mathcal{H}(\Omega)} \alpha_{w'} \sin(w' \cdot \mathbf{x}), \quad \alpha_{w'} \propto \mathbf{W}_2 \cdot \prod_{c=0}^{c_1-1} \mathcal{J}_{s_c}(\mathbf{W}_{1[\cdot,c]}) \\ \mathcal{H}(\Omega) &\subseteq \{\sum_{c=0}^{c_1-1} \beta_c \Omega_c | \beta_c \in \mathbb{Z} \ \ \& \ \ \sum_{c=0}^{c_1-1} \beta_c \leq K\}, \end{aligned} \tag{6.2}$$

where $\mathcal{J}_{s_c}$ denotes a Bessel function, $\mathbf{W}_{1[\cdot,c]}$ denotes the column c of $\mathbf{W}_1$. The equation 6.2 reveals the properties of each $\sin(\cdot)$ activated INR layer in two aspects. First, the output spectrum of layer 2, i.e. $\alpha_{w'}$, is dependent on the spectrum of layer 1, determined by $\mathbf{W}_1$; Second, the output frequency domain $\mathcal{H}(\Omega)$ is sparse since β_c is an integer, so $\mathcal{H}(\Omega)$ only covers sparse frequency space spanned by the basis $\{\beta_c \Omega_c\}$. These suggest that INR

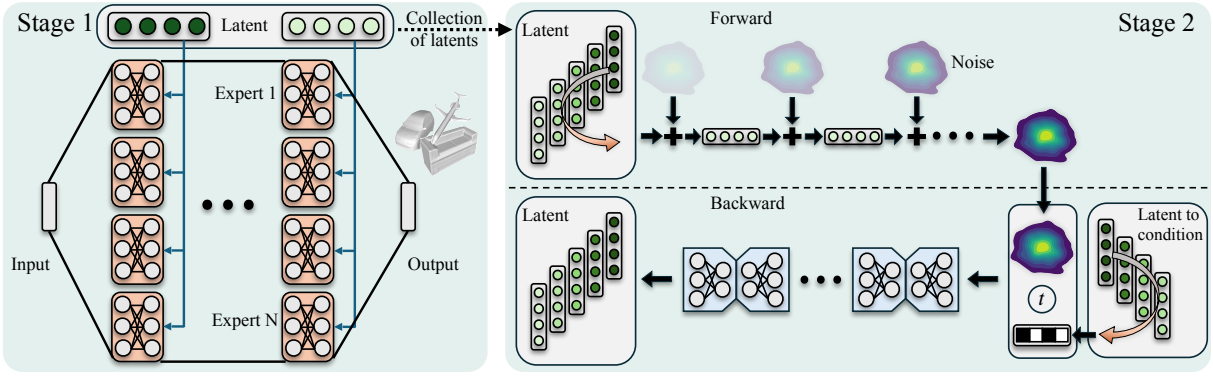


Figure 6.2: The proposed CHINR consists of two stages. In Stage-1, a Layer-of-Experts (LoE) model is used to represent large-scale data, utilizing instance-specific latents and shared experts. Stage-2 introduces a Hierarchical Controllable Diffusion Model to learn the layer-wise conditional distributions of the latents.

layers' spectrum and frequency basis inherently exhibit a sparse and hierarchical structure, encoded by $\boldsymbol{\theta}$, which extends to their representation ability. Latent-modulation approaches like mNIF [332], which model the parameter distribution $p(\boldsymbol{\theta})$ with the surrogate task of modeling the latent distribution $p(\boldsymbol{h})$, overlook the hierarchy in $\boldsymbol{h}$ transferred from $\boldsymbol{\theta}$. Ignoring this hierarchy leads to reduced expressivity and generalizability, and limited control over the generation process.

6.3 Proposed Method

Our method uses a two-stage framework to align the hierarchy of data semantic and INR's expanded representation ability, as shown in Fig. 6.2. In the first stage, we train individual INRs to fit a target dataset. In the second stage, we use generative models to learn the weight distribution for generating new data. Directly modeling the INR weight distribution brings three challenges: (1) independently trained INRs make it hard to extract shared information for distribution learning, (2) the high dimensionality of raw weights is intractable, and (3) it ignores the hierarchical structure of INRs. To address these issues, we configure the INR as a Layers-of-Experts network, where each layer contains a

set of shared expert weights and an instance-specific latent vector. As shown in the left of Fig. 6.2, the inference process builds each INR layer by layer, combining the experts with the corresponding latent vector. This structure captures shared information through the experts, simplifies distribution learning by focusing on the latents, and explicitly models the conditional dependencies within the hierarchical structure of INRs. In the following sections, we first define the LOE structure and the learning task, followed by detailed explanations of the two stages.

6.3.1 Problem Statement

Suppose an INR $f_{\boldsymbol{\theta}}$ has L layers. For layer l, we learn a collection of K cross-data shared expert weights $\boldsymbol{\theta}^l = \{\boldsymbol{\theta}^l_1, \boldsymbol{\theta}^l_2, \cdots, \boldsymbol{\theta}^l_K\}$ and a unique latent $\mathbf{h}^l \in \mathbb{R}^H$ for each data instance. At inference, the parameters for layer l in an instance-specific INR are obtained by $\overline{\boldsymbol{\theta}}^l = \sum_{n=1}^{K} \boldsymbol{\theta}^l_k \cdot \alpha^l_k$, where $[\alpha^l_1, \alpha^l_2, ..., \alpha^l_K]^\top = \boldsymbol{\alpha}^l$ denotes the *gating vector* and is computed by a gating module $\boldsymbol{\alpha}^l = g_{\boldsymbol{\phi}}(\mathbf{h}^l)$. Compared with directly learning the gating vectors, the function $g_{\boldsymbol{\phi}}(\cdot)$ allows for a more compact latent that benefits distribution learning. By modulating the contribution of experts via latents, each layer gains the flexibility to adapt to individual data samples using a shared basis. Since L layers are cascaded to form the final INR, its expressive capacity is significantly enhanced through the integrated contributions across layers.

In Stage-1, we optimize the shared network parameters denoted as

$$\boldsymbol{\theta} = \{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^L, \boldsymbol{\phi}\},$$

and fully characterize each instance-specific INR by layer-wise stacked latents $\mathbf{h} = [\mathbf{h}^1, \mathbf{h}^2, \cdots, \mathbf{h}^L] \in \mathbb{R}^{H \times L}$. The layer-wise latent structure enables hierarchical modeling of INR parameters, which aligns with the inherent hierarchy of data semantics, allowing for layer-wise dependency modeling in Stage-2, and controllable data generation.

6.3.2 Stage-1: Learning a dataset of LoE INRs

Similar to Functa [57] and mNIF [332], we use meta-learning and auto-decoding to train both the data-specific latents $\mathbf{h}$ and the shared parameters $\boldsymbol{\theta}$ for the LOE INR during Stage-1. For meta-learning, we adopt an interleaved training procedure inspired by CAVIA [375], where the experts and latents are updated alternately in separate training loops. In the inner loop, we fix $\boldsymbol{\theta}$ and adapt the latents $\mathbf{h}$ to data samples. Within each inner loop, $\mathbf{h}$ is first randomly initialized around zero and then updated for a few steps. In the outer loop, $\boldsymbol{\theta}$ is optimized based on the updated $\mathbf{h}$. This ensures each data-specific latent can be effectively learned within a few iterations, encouraging faster convergence and adaptation to new data, which is essential for distribution modeling and generalization in Stage-2. In the case of auto-decoding, we jointly optimize all parameters, maintaining a latent bank for the dataset and updating the sampled batch of latents in each iteration. Unlike meta-learning, auto-decoding does not require second-order derivatives, making it more computationally efficient. Due to this efficiency, we apply auto-decoding specifically for NeRF training.

In both approaches, each data-specific latent $\mathbf{h}$ consists of L components separately modulating the L layers in the LOE INR. This setup facilitates conditional distribution modeling for Stage-2, as opposed to learning their joint distribution like mNIF and Functa, enabling hierarchical and controllable generation.

6.3.3 Stage-2: Conditional distribution learning for generalizable INRs

Given a collection of learned latents $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots \mathbf{h}_N | \mathbf{h}_n \in \mathbb{R}^{H \times L}\}$, where N denotes the number of data instances, L the number of layers, and H the dimension of each layer of the latent, the Stage-2 aims at learning the distribution of latent $p(\mathbf{h})$. Instead of blindly modeling the joint distribution, we reformulate $p(\mathbf{h})$ as follows:

$$p(\mathbf{h}) = p(\mathbf{h}^1, \mathbf{h}^2, \cdots, \mathbf{h}^L) = p(\mathbf{h}^1) \prod_{l=2}^{L} p(\mathbf{h}^l | \mathbf{h}^{<l}), \tag{6.3}$$

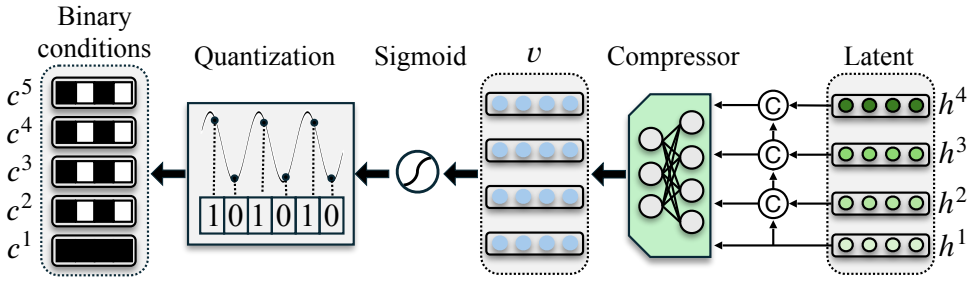


Figure 6.3: Condition formation. Layers of latent $\mathbf{h}$ are first concatenated and then compressed. The compressed tensors are binarized to generate low-dimensional binary conditions.

where $p(\mathbf{h}^{<l}) = p(\mathbf{h}^1, \cdots, \mathbf{h}^{l-1})$ denotes the joint probability of first $l-1$ layers. We design a hierarchical conditional diffusion model to learn the conditional dependency $p(\mathbf{h}^l|\mathbf{h}^{<l})$ in equation 6.3. As shown in Fig. 6.2, Stage-2 illustrates the architecture of the HCDM, which includes a forward process and a backward process.

Forward Process

We initialize the $\mathbf{h}$ at step 0 as $\mathbf{h}_0 = [\mathbf{h}_0^1, \cdots, \mathbf{h}_0^L]$ with a conditional chain of length L, the forward process for each layer $\mathbf{h}^l$ is formulated as:

$$q(\mathbf{h}_{1:T}^l|\mathbf{h}_0^l) := \prod_{t=1}^{T} q(\mathbf{h}_t^l|\mathbf{h}_{t-1}^l), \quad q(\mathbf{h}_t^l|\mathbf{h}_{t-1}^l) := \mathcal{N}(\mathbf{h}_t^l; \sqrt{1-\beta_t}\mathbf{h}_{t-1}^l, \beta_t\mathbf{I}), \tag{6.4}$$

where $q(\mathbf{h}_t^l|\mathbf{h}_{t-1}^l)$ denotes the posterior distribution of the $\mathbf{h}_t^l$ given $\mathbf{h}_{t-1}^l$, T denotes the number of diffusion steps. $\beta_1, \cdots, \beta_T$ denotes the variance schedule of the added Gaussian noise $\mathcal{N}(\cdot)$. By the forward process, the noise sample $\mathbf{h}_T = [\mathbf{h}_T^1, \cdots, \mathbf{h}_T^L]$ is generated from $\mathbf{h}_0 = [\mathbf{h}_0^1, \cdots, \mathbf{h}_0^L]$.

Backward Process

The backward process models the prior distribution defined by equation 6.3. To model the hierarchical structure, the backward process should express the conditional dependency $p(\mathbf{h}^l|\mathbf{h}^{<l})$ for all L components. Therefore, we take the $\mathbf{h}^{<l}$ as condition, and generate $\mathbf{h}^l$ for $l = 1, \cdots L$ iteratively. To understand how to process $\mathbf{h}^{<l}$ as a condition, we first explain the details of condition formation.

Condition Formation. To generate a latent $\mathbf{h} = [\mathbf{h}^1, \cdots, \mathbf{h}^L]$, the condition formation prepares for each $\mathbf{h}^l$ a condition vector $\mathbf{c}^l$ encapsulating $\mathbf{h}^{<l}$. Since $\mathbf{h}$ resides on a low-dimensional manifold (e.g., 64), the condition $\mathbf{c}^l$ should contain less information to prevent the HCDM from memorizing all one-to-one mappings from $\mathbf{h}^{<l}$ to $\mathbf{h}^l$, where $p(\mathbf{h})$ inevitably degenerates into $p(\mathbf{h}^1)$. Therefore, we embed the conditions $\mathbf{h}^{<l}$ into a lower-dimensional binary vector. Fig. 6.3 shows the framework of condition formation, which includes two steps. (1) All $\mathbf{h}^{<l}$ are concatenated and compressed with a compressor $\mathbf{W}$ to get a compressed tensor $\mathbf{v}^l \in \mathbb{R}^C$: $\mathbf{v}^l = \mathbf{W} \cdot \text{concat}(\mathbf{h}^1, \cdots, \mathbf{h}^L), \mathbf{W} \in \mathbb{R}^{C \times HL}, \mathbf{h}^j \in \mathbb{R}^H$, where C denotes the compressed dimension. To match the dimensions of $\mathbf{W}$ with the concatenated tensors, we set the values of the unused portion to zero. (2) Given the compressed tensor $\mathbf{v}^l$, a binarized condition tensor $\mathbf{c}^l$ is obtained by: $\mathbf{c}^l = \mathcal{Q}(\sigma(\mathbf{v}^l))$, where $\mathcal{Q}(\cdot)$ denotes the binarization operation, σ denotes the "Sigmoid" function. For $\mathbf{c}^1$, we set it as a zero tensor since $\mathbf{h}^1$ has no condition. Now we get the condition $\mathbf{c}^1, \cdots, \mathbf{c}^L$.

Hierarchical generation. With the condition $\mathbf{c} = [\mathbf{c}^1, \cdots, \mathbf{c}^L]$, noise sample $\mathbf{h}_T = [\mathbf{h}_T^1, \cdots, \mathbf{h}_T^L]$, time step t, we are all prepared to generate a complete latent $\mathbf{h}_0$. To generate a component $\mathbf{h}_0^l$, the backward process is formulated as: $p(\mathbf{h}_{0:T}^l|\mathbf{c}^l) = p(\mathbf{h}_T^l) \prod_{t=1}^T p(\mathbf{h}_{t-1}^l|\mathbf{h}_t^l, \mathbf{c}^l)$. It is noted that a complete backward process, generating a sample from $p(\mathbf{h}_0^l|\mathbf{c}^l)$, is exactly the implementation of $p(\mathbf{h}^l|\mathbf{h}^{<l})$ in equation 6.3, where $\mathbf{c}^l$ corresponds to $\mathbf{h}^{<l}$. We adopt a UNet [236] $\mu_{\boldsymbol{\theta}}$ as in [260, 104] to implement $p(\mathbf{h}_{t-1}^l|\mathbf{h}_t^l, \mathbf{c}^l)$:

$$
\begin{aligned}
&p(\mathbf{h}_{t-1}^l|\mathbf{h}_t^l, \mathbf{c}^l) = \mathcal{N}(\mathbf{h}_{t-1}^l : \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{h}_t^l, t, \mathbf{c}^l), \boldsymbol{\Sigma}(t)) \\
&\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{h}_t^l, t, \mathbf{c}^l) = \frac{\sqrt{a_t}(1-\bar{a}_{t-1})\mathbf{h}_t^l + \sqrt{\bar{a}_{t-1}(1-a_t)}\mu_{\boldsymbol{\theta}}(\mathbf{h}_t^l, t, \mathbf{c}^l)}{1-\bar{a}_t} \\
&a_t = 1 - \beta_t, \quad \bar{a}_t = \prod_{i=1}^{t} a_i, \quad \boldsymbol{\Sigma}(t) = \frac{(1-a_t)(1-\bar{a}_{t-1})}{1-\bar{a}_t}\mathbf{I}.
\end{aligned}
\tag{6.5}
$$

Each backward process starts from a Gaussian noise sample $\mathbf{h}_T^l$ and generates a latent component $\mathbf{h}_0^l$ with condition $\mathbf{c}^l$. By iteratively sampling from $p(\mathbf{h}^1)$ and $p(\mathbf{h}^l|\mathbf{h}^{<l})$ with $l = 2, \cdots L$, a complete latent $\mathbf{h}_0 = [\mathbf{h}_0^1, \cdots, \mathbf{h}_0^L]$ is generated. The final objective is: $\mathcal{L}_{hcdm} = \mathbb{E}_{\mathcal{H},t,l}[\lambda||\epsilon, \epsilon_{\boldsymbol{\theta}}(\mathbf{h}_t^l, t, \mathbf{c}^l)||^2]$, where $\mathcal{H}$ denotes all latents, ϵ denotes Gaussian sample from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and λ is a constant coefficient. For inference, we first generate $\mathbf{h}^1$ from Gaussian noise, then perform a chain of conditional sampling from $p(\mathbf{h}^l|\mathbf{h}^{<l})$ until we get the complete latent $\mathbf{h}$ used by the LOE to generate new data.

6.4 Experiments

In this section, we begin by introducing the datasets and evaluation criteria. Then, we provide quantitative and qualitative results to demonstrate the efficacy of our proposed framework. Next, we highlight the controllability of the INR generation process, enabled by conditional dependency modeling and hierarchical sampling. Furthermore, we analyze the latent space and illustrate how data semantics are embedded within INR's weight space hierarchically. We show that this structure grants the INR a *compositional* property across layers. Finally, we conduct ablation studies to show the importance of conditional dependency modeling and the functionality of binary conditions.

6.4.1 Details on Experimental Setup

Implementation details. The LOE structure can be configured with the number of layers L, the number of experts at each layer K, the channel dimension of each expert C, and the dimension of the latent at each layer H, denoted as a tuple (L, K, C, H). We train LOEs of $(7, 384, 128, 128)$, $(5, 256, 64, 256)$, $(6, 256, 64, 64)$, and $(5, 64, 64, 64)$ in CelebA-HQ [118], ShapeNet [23], SRN-Cars [259], and AMASS [178] datasets, respectively. We follow mNIF [332] on the data processing protocols for CelebA-HQ, ShapeNet, and SRN-Cars datasets. Details about the AMASS dataset are provided in Sec. 6.5.3.

Training details. In our experiments, both Stage-1 and Stage-2 are trained and evaluated on the CelebA-HQ 64^2 [118], ShapeNet 64^3 [23], SRN-Cars [259], and AMASS [178]

datasets. All experiments are implemented in Pytorch and run on a single Nvidia RTX3090 GPU. For evaluation metrics, we use peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [296], and accuracy to assess Stage-1 reconstruction, and Fréchet inception distance (FID) [103], coverage, and maximum mean discrepancy (MMD) [4] to evaluate Stage-2 generation performance. In Stage-1, we train LOEs via meta-learning on CelebA-HQ, ShapeNet, and AMASS, and with auto-decoding on SRN-Cars. We use a batch size of 32, an outer learning rate of 1e−4, an inner learning rate of 1 with 3 steps, and train the LOE for 800 epochs in the meta-learning setting. For auto-decoding experiments on SRN-Cars, we use a batch size of 8, a learning rate of 1e−4, and train the LOE for 1000 epochs. In both settings, we use the AdamW [173] optimizer without weight decay. In Stage-2, we set the training batch size to be 32, learning rate 1e−4, and cosine scheduler with minimum learning rate 0.0. We train the HCDM for 1000 epochs with the AdamW optimizer.

6.4.2 Quantitative and Qualitative Results

Table 6.1 presents the reconstruction and generation performance of the proposed CHINR compared with baseline methods on different modalities. Our model outperforms existing methods on most datasets. In Stage-1, it achieves the highest reconstruction PSNR, thanks to the expanded representation capacity of LOE and layer-wise latent learning. In Stage-2, it demonstrates superior generalization, highlighting the effectiveness of hierarchical conditional modeling in capturing the diverse data semantics. On CelebA-HQ, our model achieves the second-lowest FID score, slightly behind mNIF. However, we observe that mNIF tends to generate images resembling those from the training set. To verify this, we perform retrieval for $1,000$ generated samples and compute the average of their minimum L2 distance to the training set images. A lower value indicates a higher chance of "memorizing" the training set. mNIF obtains a value of 6.243 whereas CHINR obtains 15.971, indicating that our model generalizes better by generating new images that differ more noticeably from the training data. Examples of retrieval results are provided in the Appendix.

Fig. 6.4 displays uncurated samples generated by our model compared to existing meth-

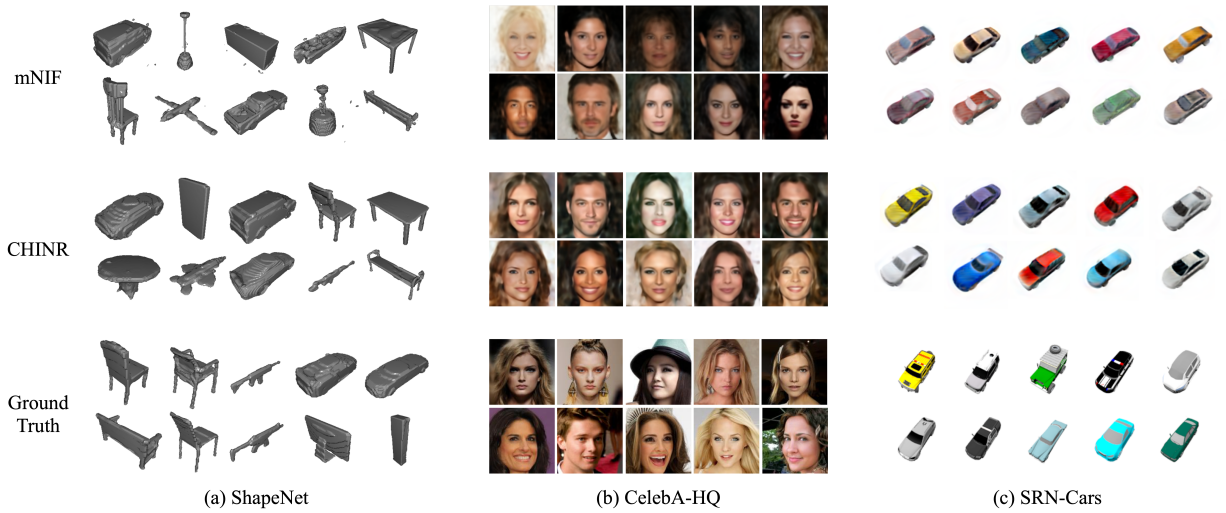


Figure 6.4: Uncurated generations for ShapeNet, CelebA-HQ, and SRN-Cars datasets.

ods. With the proposed HCDM, our model generates high-quality samples with rich details. More results including the AMASS dataset are provided in the Appendix.

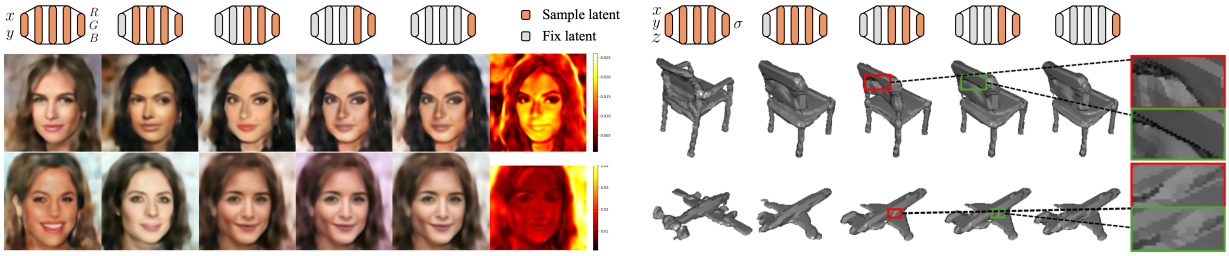


Figure 6.5: Controllable hierarchical generation by progressively fixing the layer-wise latent.

6.4.3 Hierarchical Controllable Generation

In this section, we show the hierarchical controllable generation with conditional sampling. Since the proposed HCDM successfully models the conditional dependencies of the modulation latents, we can conduct a chain of conditional samplings to achieve hierarchical

Model	CelebA-HQ			ShapeNet				SRN-Cars		
	Reconstruction		Generation	Reconstruction		Generation		Reconstruction		Generation
	PSNR↑	SSIM↑	FID↓	PSNR↑	Accuracy↑	Coverage↑	MMD↓	PSNR↑	SSIM↑	FID↓
Functa [57]	26.6	0.801	40.4	-	-	-	-	24.2	0.739	80.3
GEM [54]	-	-	30.4	21.3	-	0.409	0.0014	-	-	-
GASP [59]	-	-	13.5	16.5	-	0.341	0.0021	-	-	-
mNIF [332]	34.5	0.958	13.2	21.4	0.972	0.437	0.0013	25.9	0.758	79.5
CHINR	34.9	0.963	13.4	22.3	0.988	0.441	0.0011	26.2	0.772	77.9

Table 6.1: Quantitative results on CelebA-HQ, ShapeNet, and SRN-Cars.

Model	CelebA-HQ					ShapeNet					SRN-Cars				
	$\mathbf{std}_1$	$\mathbf{std}_{12}$	$\mathbf{std}_{23}$	$\mathbf{std}_{34}$	$\mathbf{std}_{45}$	$\mathbf{std}_1$	$\mathbf{std}_{12}$	$\mathbf{std}_{23}$	$\mathbf{std}_{34}$	$\mathbf{std}_{45}$	$\mathbf{std}_1$	$\mathbf{std}_{12}$	$\mathbf{std}_{23}$	$\mathbf{std}_{34}$	$\mathbf{std}_{45}$
$HCDM_8$	0.7766	0.9153	0.9363	0.8772	0.8631	0.8533	0.9510	0.8933	0.7682	0.7301	0.7223	1.1270	1.1108	0.9474	1.1210
$HCDM_{12}$	0.7766	0.9092	0.9292	0.8663	0.8550	0.8533	0.9424	0.8712	0.7327	0.7110	0.7223	1.0663	1.0180	0.9331	1.1803
$HCDM_{15}$	0.7766	0.5344	0.5478	0.3257	0.2570	0.8533	0.6125	0.5241	0.4088	0.3857	0.7223	0.6348	0.6432	0.5837	0.6578
$HCDM_{20}$	0.7766	0.1032	0.1121	0.0853	0.0766	0.8533	0.1051	0.0823	0.0715	0.0522	0.7223	0.1048	0.1122	0.0821	0.0933

Table 6.2: Ablation study of different binary condition lengths, 8, 12, 15, and 20, when training Stage-2 on CelebA-HQ, ShapeNet, and SRN-Cars datasets.

control over semantics at different granularities in the generated contents. Specifically, to sample a latent $\mathbf{h} = [\mathbf{h}^1, \cdots, \mathbf{h}^5]$, we begin by sampling $\mathbf{h}^1$ with HCDM taking a Gaussian noise as input. Next, we sample $\mathbf{h}^2$, conditioned on the binary vector generated by $\mathbf{h}^1$. This process continues layer by layer until $\mathbf{h}^5$ is generated, forming a complete latent $\mathbf{h}$. This layer-wise latent then modulates the LOE to render output such as images, point clouds, or NeRF renderings. Fig. 6.5 illustrates this hierarchical control on two modalities. For each modality, we show two samples generated through the full chain of conditional sampling ($\mathbf{h}^1 \cdots \mathbf{h}^5$) in the first column. In the second column, we fix their $\mathbf{h}^1$ from the first column and sample the remaining parts ($\mathbf{h}^2 \cdots \mathbf{h}^5$). In the third column, we fix both $\mathbf{h}^1$ and $\mathbf{h}^2$ as in the second column and conditionally sample the rest. This progressive fixation allows us to control the finer details of the generated output.

As shown Fig. 6.5, two different chains of conditional sampling, each starting from a different initial $\mathbf{h}^1$, result in highly different data semantics (first column). For the facial images, when $\mathbf{h}^1$ is fixed, the generated samples exhibit similar overall contours but different contents, i.e., facial features, facial orientation, and hairstyles (second column). When $\mathbf{h}^1$ and $\mathbf{h}^2$ are fixed, variations occur primarily in facial details, such as eye shape and hair color (third column). Fixing the first three parts (fourth column) results in changes limited to skin tone. Lastly, $\mathbf{h}^5$ influences global properties like foreground or background color, as highlighted by the heat maps in the last column. Similarly, for voxels, the hierarchy control is manifested in category (e.g., chair and plane), object parts (e.g., armrest and back of the chair), and finer details of specific parts (e.g., structure of the chair back).

This hierarchical control via conditional sampling demonstrates that HCDM effectively captures conditional dependencies of parameters, aligned with the hierarchy of data semantics. It also offers practical benefits in digital content creation, allowing creators to fine-tune specific details in their work. Additional examples of controllable generation are provided in the Appendix.

6.4.4 Analysis

In this section, we unravel the reason behind the success of hierarchical controllable generation. We show that each layer's latent modulates disentangled semantics, giving our

LOE model a compositional property. Moreover, the layer-wise conditional dependency modeling successfully captures the hierarchy of data semantics, ensuring the composed semantics are compatible across layers.

Latent Composition

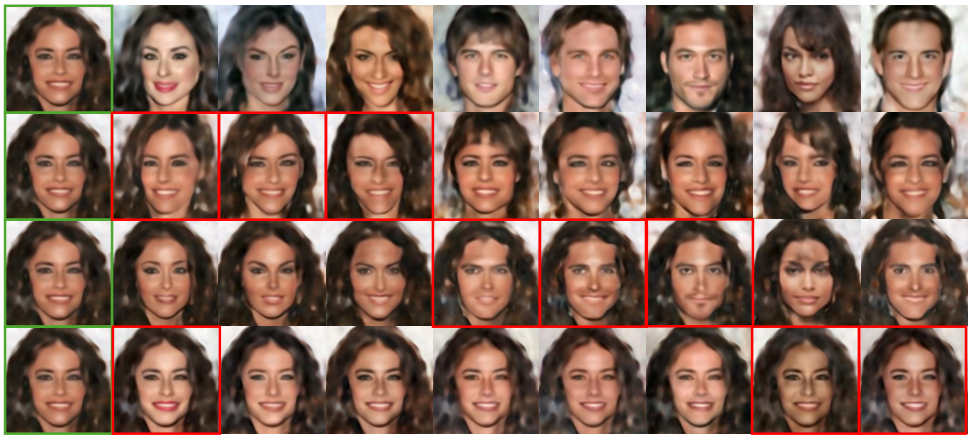


Figure 6.6: Latent composition. The first row presents 9 randomly sampled images from Stage-2. The second to the fourth rows present the images where the second to the fourth parts of the first sample's latent (green boxes) are replaced. The representative examples are highlighted in red boxes.

We explore the controllability of our method through latent composition and find that the learned latents are layer-wise compositional. Given two latents $\mathbf{h}_1 = [\mathbf{h}_1^1, \cdots, \mathbf{h}_1^5], \mathbf{h}_2 = [\mathbf{h}_2^1, \cdots, \mathbf{h}_2^5]$, exchanging a specific part, e.g. $\mathbf{h}_1^2$ and $\mathbf{h}_2^2$, results in the corresponding semantic changes in the generated content. As shown in Fig. 6.6, we randomly sample 9 latents from HCDM, trained on CelebA-HQ, and render the images in the first row. These faces exhibit diverse characteristics including expressions, hairstyles, facial orientations, skin tones, and background colors. In the second row, we replace the second layer's latent of the first sample with that of the other 8 samples, while keeping the rest latents unchanged. It's clear that the facial orientation, hairstyles, and background colors change accordingly, while the facial features remain the same. This indicates that the second part

of the latent encodes these specific semantics disentangled from other semantics, laying the foundation for the controllable hierarchical generation. In the third row, when the third layer's latent is replaced, we observe that only the facial features are swapped, while other characteristics, such as facial orientation, remain unchanged. In the last row, only skin tone changes.

It's important to note that latent composition disrupts the conditional chain, meaning the newly composed latents may be incompatible across layers, resulting in lower-quality images. Nevertheless, we conduct latent composition to illustrate how and why our method works. The results show that the image semantics can be embedded and disentangled in parameter space, offering a new perspective on image generation.

Layer-wise Hierarchy Analysis

Layer-wise correlation. We perform layer-wise correlation analysis on the latents to show the necessity of conditional dependency modeling. We compute the cross-layer correlation between latents using Singular Vector Canonical Correlation Analysis (SVCCA) [223], a metric that measures the correlation between neural network representations. Fig. 6.7 (a) displays the pairwise correlations between $\mathbf{h}$ across layers, trained on Celeba-HQ in Stage-1, showing the non-negligible correlation between layers. This underscores the importance of modeling conditional distributions $p(\mathbf{h}^l|\mathbf{h}^{<l})$, rather than independent marginal distributions $p(\mathbf{h}^l)$ in Stage-2. More results on other datasets can be found in the Appendix. We next show the learned hierarchical structures of latents.

Layer-wise dependency visualization. We visualize the hierarchical dependencies of latents to better understand how data semantics are encoded in INR's weight space, as the latents are mapped to gating vectors that modulate the experts at each layer. Fig. 6.7 (b) and (c) show the conditional distributions of latents from adjacent layers, trained on CelebA-HQ. Specifically, we apply PCA to the latents and plot their distributions in grayscale. For example, the Fig. 6.7 (b) shows distributions for layers 1 and 2. We then run the generation process five times to obtain five sampled latents at each layer, as depicted in color in the left parts of Fig. 6.7 (b) and (c). Based on the HCDM, we can plot the resulting conditional distributions of latents at each subsequent layer, represented

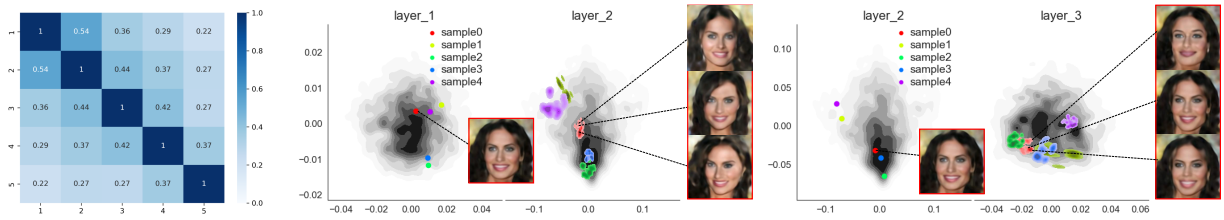


Figure 6.7: Layer-wise hierarchy analysis. (a): correlation between cross-layer latents, obtained on the train-split of CelebA-HQ at Stage-1. (b) and (c): visualization of conditional distributions across layers. The gray regions show the distribution of latents from Stage-1, while the colored regions represent the sampled latents from Stage-2.

by colored regions in the right parts. Additionally, we show the final generated images corresponding to different samples. We can see clear patterns of a hierarchical structure, which corresponds to the semantic variations at different granularities. For example, the sampled latents in layer 1 determine the overall contours of the face. When layer 2 is determined, variations in layer 3 modify the facial expressions while keeping the orientation fixed. In fact, different layers govern different aspects of the generated data, as we have seen in the compositional analysis. Furthermore, the latent sampling space at each layer is constrained by the preceding layers, ensuring compatibility between layers in representing the data. Therefore, the generated data semantics are hierarchically controllable.

6.4.5 Ablation Studies

Ablation on condition modeling. To demonstrate the importance of conditional dependency modeling for hierarchical controllable generation, we train an unconditional diffusion model that directly maps noise to layer-wise latents in Stage-2. We then sample $p(\mathbf{h}^l)$ independently for $l = 1, \cdots, L$ to generate the full latents. The resulting images for CelebA-HQ are shown in Fig. 6.8 (a), while images generated with conditional modeling are shown in (b) for comparison. Although human faces are recognizable in (a), the noticeable artifacts highlight that independently sampled layer-wise latents fail to ensure consistent semantic composition across layers. In contrast, conditional modeling successfully achieves this compatibility.



(a) Without condition (b) With condition

Figure 6.8: Ablations on conditional modeling.

Ablation on binary condition. We demonstrate that the length of binary conditions impacts the effectiveness of learning conditional dependencies, as shown in Table 6.2. We set the binary lengths to $8, 12, 15, 20$ and train the HCDM on different datasets. Initially, we sample 5000 latents and compute the standard deviation of the first part, denoted as $\mathbf{std}_1$. Since the first part has no conditions and is sampled from noises, it shows high values irrelevant to the binary lengths. Subsequently, we select 10 random samples from the first part to use as conditions and get 5000 samples for the second part. Here, the standard deviation, denoted as $\mathbf{std}_2$, decreases as the binary condition length increases, because longer binary conditions contain more information from preceding layers. Once the length reaches a certain threshold, the standard deviation approaches zero, turning the conditional chain into a direct one-to-one mapping, thus diminishing controllability. However, if the length becomes too small, e.g. 0, all parts will be independent thus losing conditional dependency. Therefore, we empirically set the length to be 12 for Stage-2 training. We repeat this procedure for other parts and observe similar results.

Model	MSE↓
mNIF [332]	0.015
CHINR	0.011

Table 6.3: Quantitative results on AMASS.

6.5 Additional Experimental Results

6.5.1 Generalizability Analysis Through Retrieval

We use retrieval to compare the generalizability of CHINR and mNIF on the CelebA-HQ dataset. Specifically, we generate samples and retrieve the closest images from the training set. As shown in Fig. 6.9, mNIF generates samples that are very similar to the training images, suggesting a higher chance of "memorization". In contrast, CHINR demonstrates better generalization by producing "new" samples that differ more noticeably from the training data.

6.5.2 More Generated Samples

Fig. 6.10 shows more generated samples on CelebA-Net, ShapeNet, and SRN-Cars datasets.

6.5.3 AMASS Experiments

We apply our proposed CHINR model to the AMASS dataset of 3D human motions. For each motion sequence, we use 200 frames, with each frame represented by 165 values corresponding to the locations and rotations of body joints. As a result, each data instance is formatted as a grid with size 200×165. In Stage-1, the LOE is employed to fit the motion instances. In Stage-2, we set the binary lengths to 8 to avoid memorizing conditions.

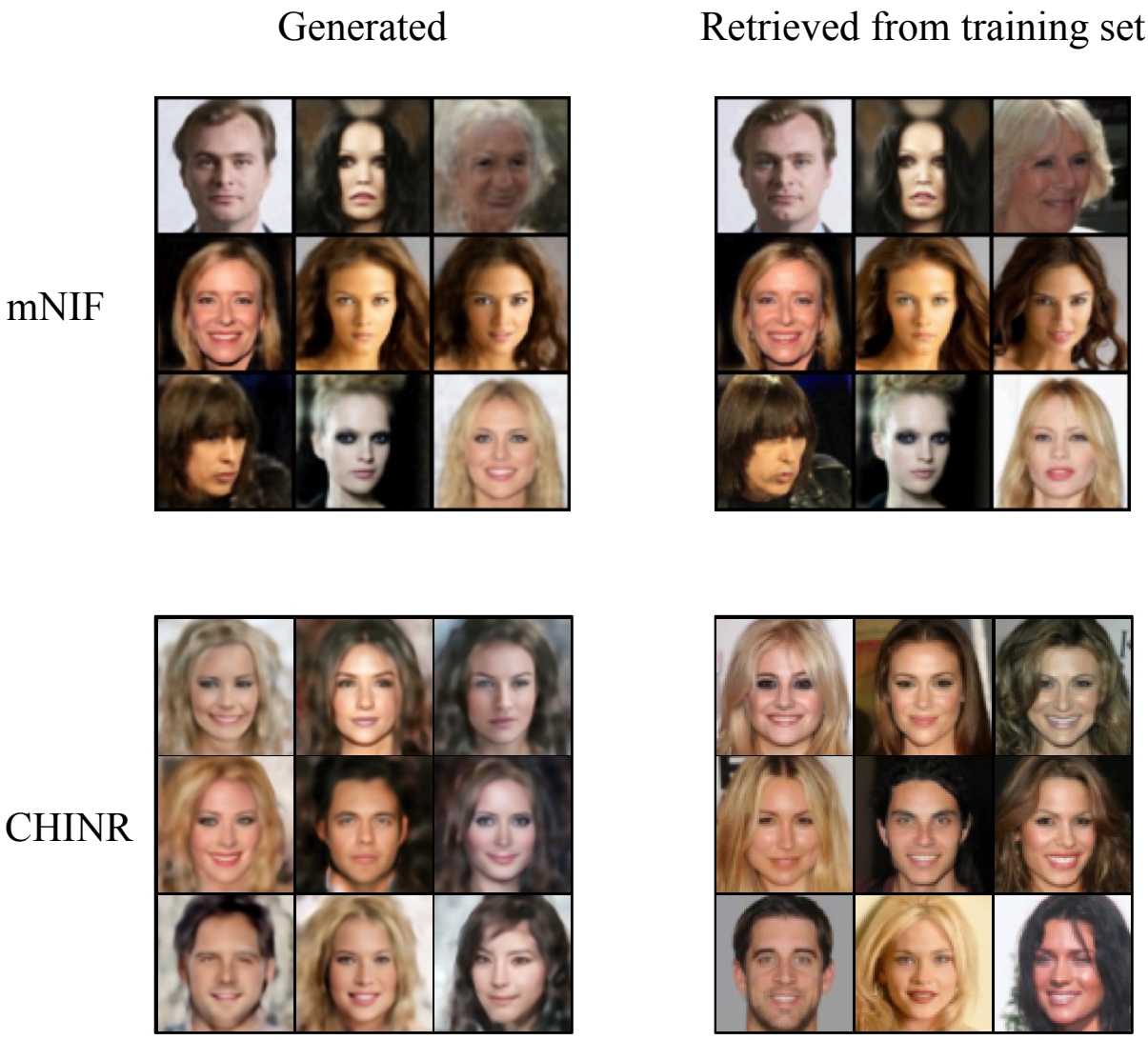


Figure 6.9: Retrieval on CelebA-HQ: mNIF retrieves images closely resembling those from the training set, while CHINR demonstrates better generalization by producing distinct new images.

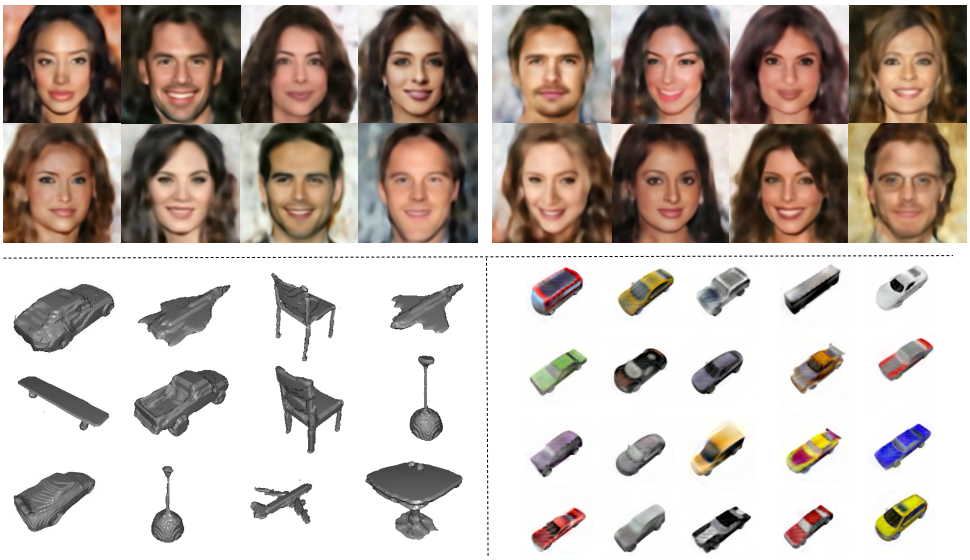


Figure 6.10: More generated samples of CelebA-HQ, ShapeNet, and SRN-Cars data.

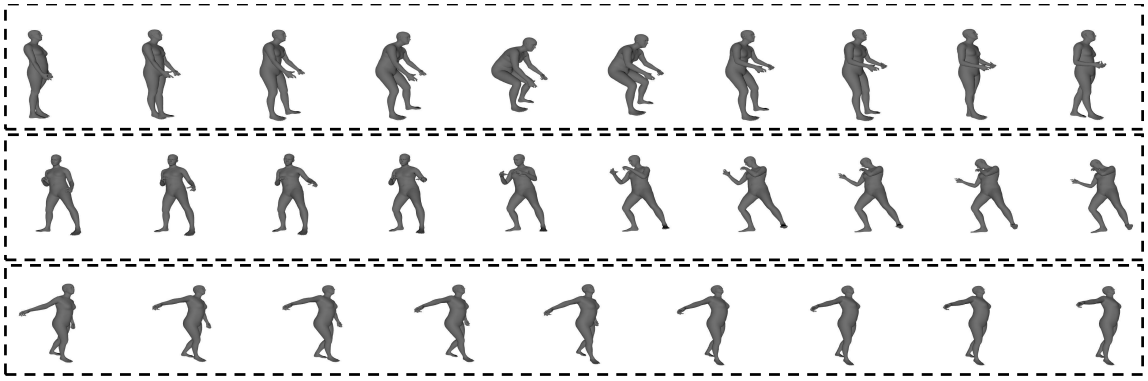


Figure 6.11: Generated motions with HCDM. each row denotes one sampled data.

Reconstruction and generation results. The reconstruction performance is shown in Table. **??**. The randomly generated motions are shown in Fig. 6.11.

Semantic-level Interpolation. Since the LOE successfully learns the consistent latent space, we can perform semantic-level interpolation for motions. As shown in Fig. 6.12, given two fitted sequential motions with LOE, each corresponds to a latent, we can interpolate the latent from the start motion (indicated by the red dashed box) to the end

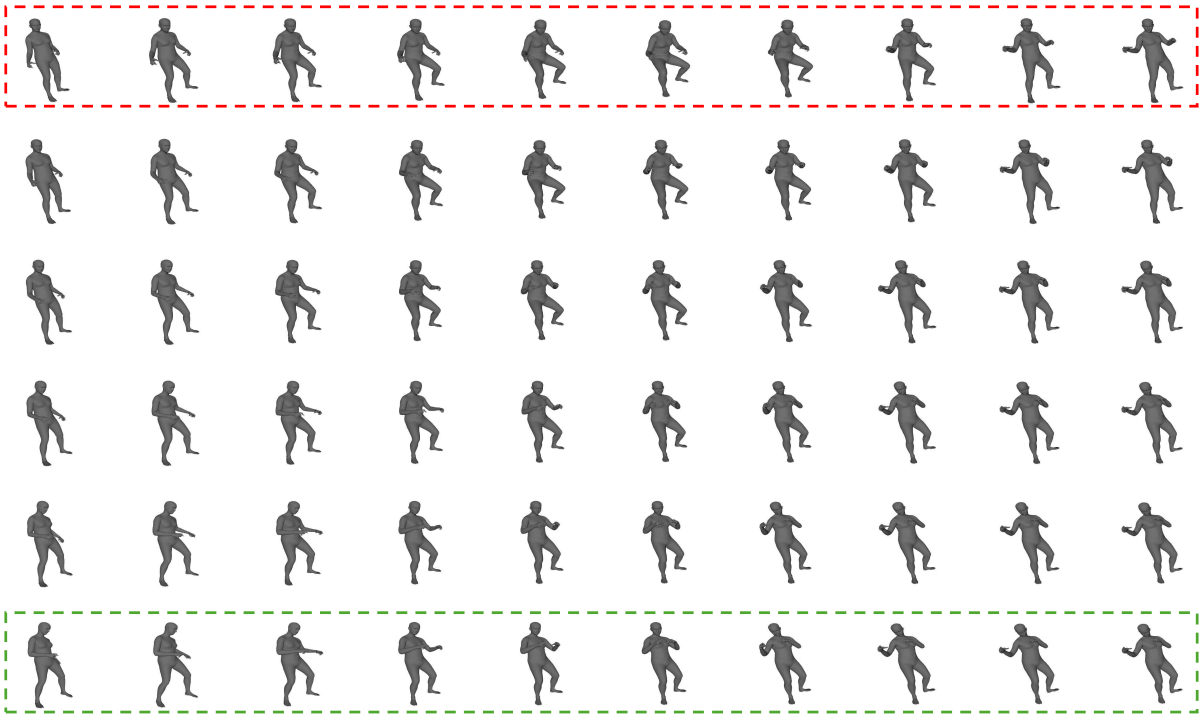


Figure 6.12: Semantic interpolation for AMASS data. Anchor sequential motions (indicated by the red and green dashed boxes) are first fitted with LOE to obtain latents. Then semantic-level interpolation is performed by interpolating the latents. The red dashed box denotes the start motion, and the green dashed box denotes the end motion.

motion (indicated by the green dashed box) linearly with ratio $[0.2, 0.4, 0.6, 0.8]$. We can see that the interpolated motions change smoothly from the start to the end. Semantic-level interpolation can be useful in the gaming industry, and 3D-digital content generation.

6.5.4 Hierarchical Controllable Generation

More examples of hierarchical controllable data generation are presented in Fig 6.13.

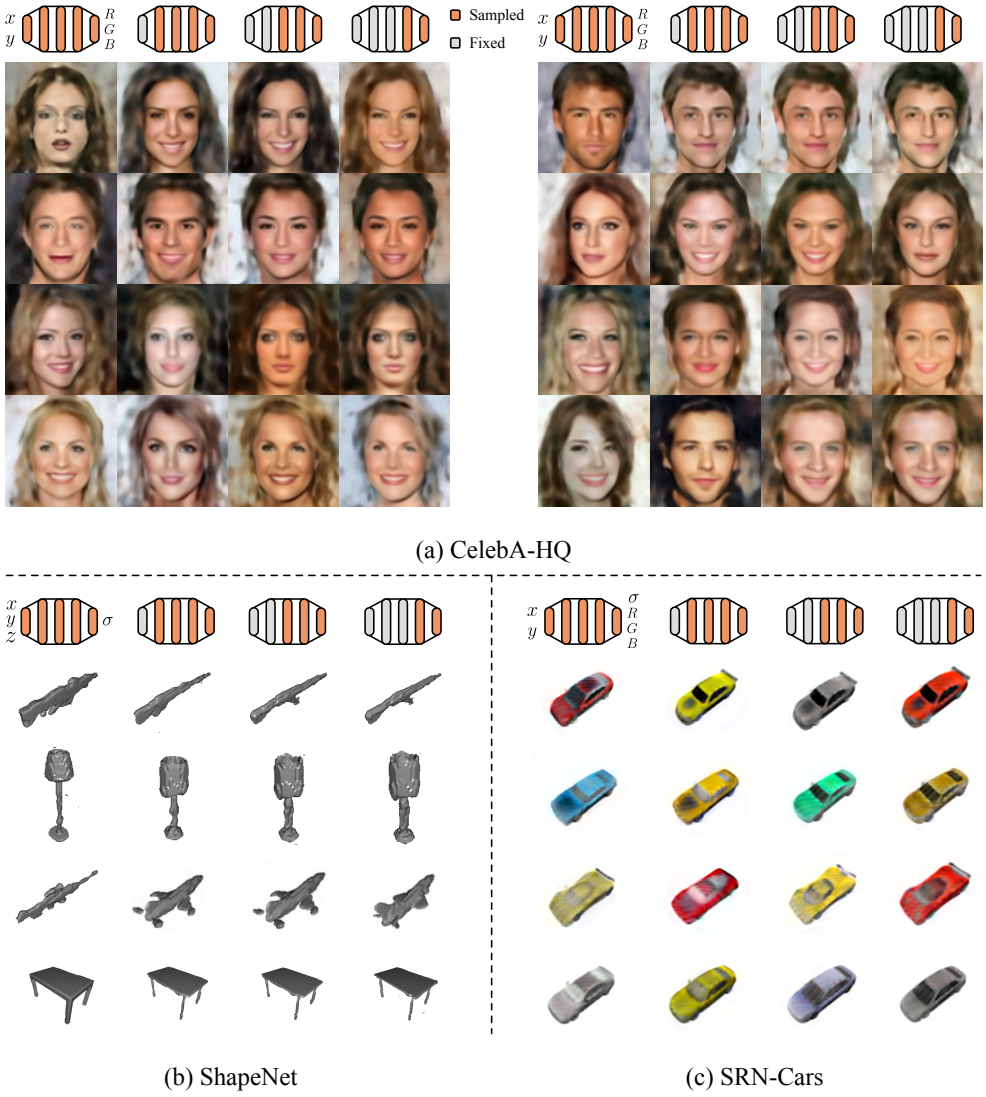


Figure 6.13: More examples of hierarchical controllable generation on CelebA-HQ, ShapeNet, and SRN-Cars data.

6.5.5 Latent-based Retrieval

We show an application of data retrieval by latents, since they already embed rich semantic meanings. We first obtain the latents for the target data by fitting it to the LOE through

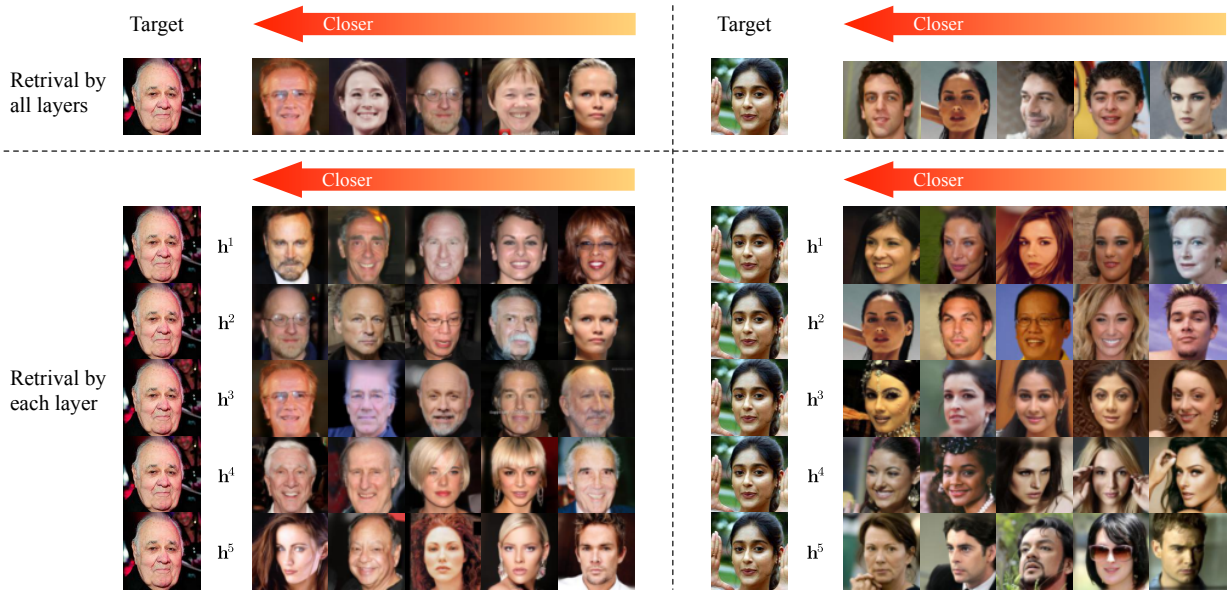


Figure 6.14: Latent-based retrieval via two approaches: retrieval by all layers and retrieval by each layer.

a few gradient steps. Once the latents are optimized, they can be used to retrieve similar data by comparing their latent representations to the searched set, allowing us to search for semantically similar examples within the latent space. Fig. 6.14 shows this process by using images from the test-split of CelebA-HQ as the targets, and train-split images as the searched set. We demonstrate two approaches for retrieval: (1) using the flattened $\mathbf{h}$ for all layers, and (2) layer-wise retrieval using each layer's latent $\mathbf{h}^l$. As shown in Fig. 6.14, retrieval by the flattened $\mathbf{h}$ will retrieve samples that are broadly similar, while layer-wise retrieval retrieves samples with specific semantic similarities. For example, $\mathbf{h}^2$ retrieves faces with similar orientations, while $\mathbf{h}^3$ retrieves faces with similar facial features such as eye shape and expressions.

6.6 Additional Analysis

In this section, we provide more analysis of the latent space and the functionalities of binary conditions.

6.6.1 Latent Space Analysis

Here, we analyze the latent space further, focusing on its interpolation capabilities and providing additional results of correlation analysis.

Latent Interpolation

To illustrate that our model learns a consistent and metric latent space, following definitions in [54], we perform latent space interpolation in two ways: complete interpolation, and layer-wise interpolation.

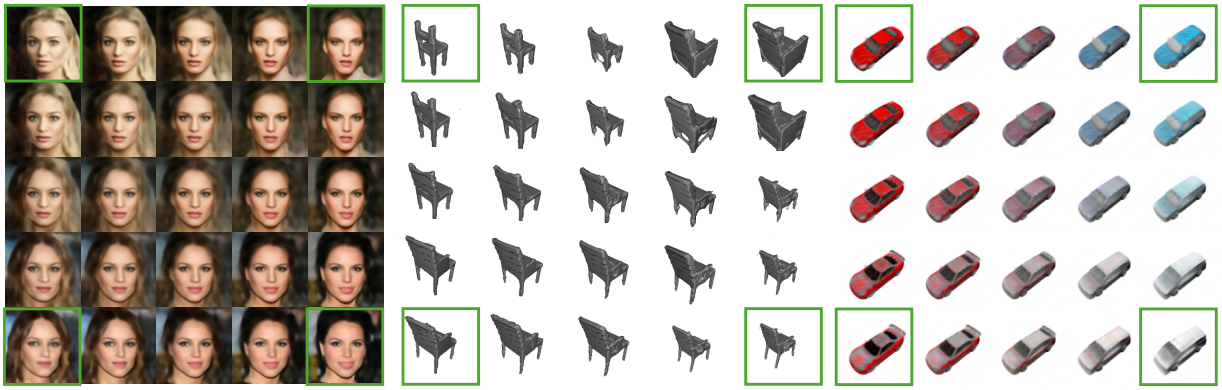


Figure 6.15: Latent space interpolation is performed for LoE, with four corner points representing the anchor examples rendered in stage 1. The intermediary points are generated through the bilinear interpolation of the latents associated with these four anchors. The interpolation is evaluated on datasets CelebA-HQ, ShapeNet, and SRN-Cars.

Complete Interpolation is shown in Fig.6.15. Four corners present the signals with latent generated from Stage-1. The intermediary signals are bilinearly interpolated from

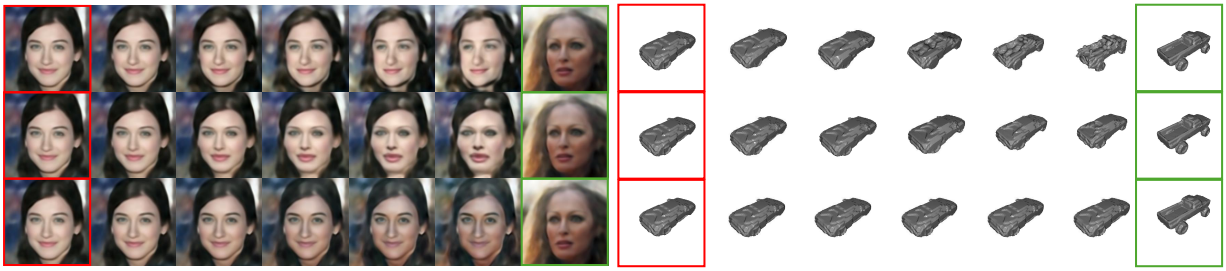


Figure 6.16: Layerwise interpolation. The red boxes denote the start and the green boxes denote the end. For the CelebA-HQ, the layers $2 \rightarrow 4$ are interpolated respectively while other layers are fixed. For the ShapeNet, the layers $1 \rightarrow 3$ are interpolated respectively.

four corners in latent space. The results demonstrate that the learned latent is metric and consistent with human perception.

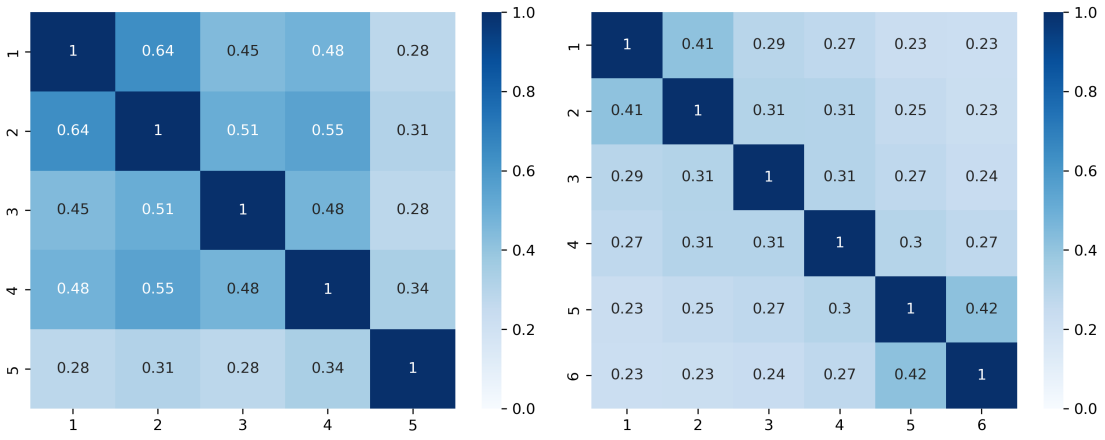
Layer-wise Interpolation. Since our LOE embeds semantics hierarchically in different parts of the latent, we can interpolate each part to control specific semantics. As shown in Fig. 6.16, we interpolate the second, third, and fourth parts of the latent associated with red-boxed signals, with the corresponding parts of the right side latent. For CelebA-HQ samples, we find that the facial orientation, facial features, and skin tone can be interpolated independently. This demonstrates that each part of the latent also constructs a metric and consistent manifold.

Layer-wise Correlation Analysis

We provide correlation analysis on additional datasets in Fig. 6.17 and Fig. 6.18. This highlights the significance of conditional modeling in the hierarchical generation process.

6.6.2 Binary condition Analysis

We analyze the clustering of latents and binary conditions on CelebA-HQ dataset, as shown in Fig. 6.19. Firstly, we use the KMeans algorithm to get 10 clusters of latents, shown as the dots in the figure. Then we select three anchor latents, generate three binary conditions with HCDM, and search the nearest binary-corresponded latents. The nearest neighbors



(a) ShapeNet. (b) SRN-Cars.

Figure 6.17: Correlation between the learned latents across layers, trained on ShapeNet [23] and SRN-Cars [259]. The non-negligible correlation between adjacent layers (e.g., $\mathbf{h}^1$ and $\mathbf{h}^2$) reveals the necessity of conditional distribution learning.

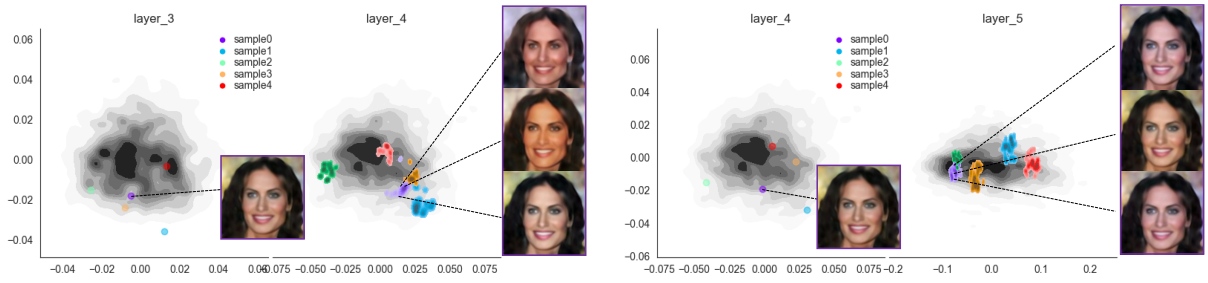


Figure 6.18: Visualization of conditional distributions across layers $3, 4, 5$. The gray regions present the distribution of latents from Stage-1, while the colored regions represent the sampled latents from Stage-2.

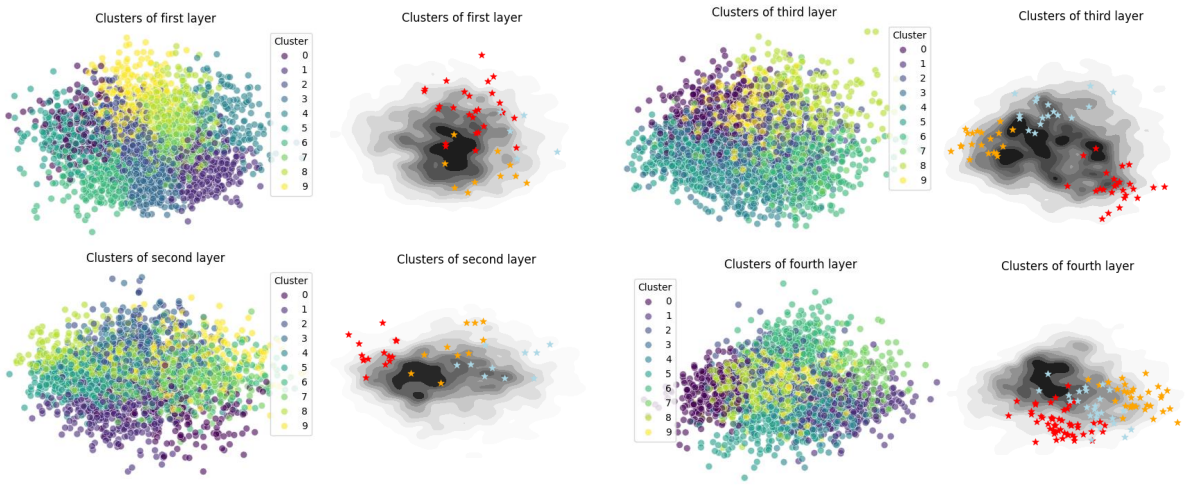


Figure 6.19: Clusters of each part of latent and binary conditions. The dotted plot presents clusters of each part of latents trained on ClebA-HQ. The gray distribution plot presents the distribution of each part of latents, and starred scatter plot presents clusters of latents with similar binary conditions.

are represented by the stars. We can observe that the binary conditions embed the latents' information and form a consistent binary condition space. This binary condition space corresponds to the latent space.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis addresses critical challenges in improving monocular depth estimation, novel view synthesis, and efficient 3D data generation by leveraging perceptual relationships, human preferences, and hierarchical representations. Each proposed framework demonstrates significant advancements in the respective domain, achieving high-quality and efficient 3D modeling. The methodologies not only advance the state-of-the-art but also provide practical 3D modeling tools for researchers.

The first work proposes a VRS framework to extract perceptual relationship representations from 2D images to improve 3D understanding. The extracted relationship representations provide global 3D constraints, which indicate the depth relations across objects. The global 3D constraints solve the problems of existing works, which rely either on low-level or local semantic features. The framework is able to identify effective relationships and quantify their contributions. Our methods achieve great performances by evaluating the framework on different depth estimation data such as KITTI and NYUv2. Further, any state-of-the-art method can be taken as a plug-in module to be equipped with relationship understanding, thus improving the accuracy. We also evaluated our method on zero-shot settings and found that the framework predicts accurate depth even when the scene is unseen.

The second work proposes a VRT framework to extract robust relationship representations to improve NVS. We make the first attempt in NVS community to integrate relationship understanding, which provides global geometry constraints, thus generating more accurate 3D modeling. Specifically, the framework can predict the perceptual relationships in novel views, which also contains a correspondence search algorithm to find the matched pair between predictions and ground-truth. Besides, the method also contains a presentness prediction module to handle different relationship variations. Three objective functions are then designed to encourage the accuracy of the predicted relationship, presentness, and structure consistency. We evaluate the accuracy of relationship prediction on different view transformations and the results demonstrate it is robust to small and large view transformation.

The third work applies human perceptual preference to improve novel view synthesis in a few-shot setting. Since the few-shot training of NeRF introduces different distortions such as floaters, quality fluctuation, and distant view degradation, we rely on human perceptual quality to help identify the distorted views and apply it to guide the training of the NeRF. To solve the problems, we propose a HuPPO framework, which utilizes an IQA model for candidate selection, a hyper-restoration module to generate positive samples, and an adversarial generation module to generate adversarial samples. Besides, we propose a HuPPO loss to formulate the quality score into an objective function. Finally, we adopt a three-step optimization strategy to train the model. The framework is model-agnostic, where any NeRF model can be inserted. We compare with different SOTA methods and achieve better performances. We further demonstrate the proposed framework can alleviate floaters, distant view degradation, and quality fluctuation.

In the last part, we propose a HCDM framework as an efficient and general representation to generate 3D data. The hierarchical modeling in the HCDM also enables controllable generation. Specifically, we propose a two-stage framework. In the first stage, a LOE model is designed to learn shared features of data by an INR model and learn distinct features by learning a latent vector. Further, each latent vector is separated into several parts to align the hierarchy of model architecture and the data semantics. In the second stage, we design a HCDM to learn the distribution of the collected latent vector. The HCDM can generate new latent vectors iteratively. The hierarchical modeling also enables controllable

generation. In the experiment, we demonstrate our method outperforms SOTA methods on both 2D images, 3D point clouds, and NeRF. We also demonstrate the methods can control the details of the generated data progressively. Further, by analyzing the information embedded in the latent vector, we find that the data semantics are disentangled in different layers of LOE.

7.2 Future work

Visual relationships define how objects interact or relate in an image or scene. For example, "A cat sitting on a chair" describes a spatial relationship. "A person holding a cup" defines an action-based relationship. Capturing these relationships enhances semantic understanding and spatial reasoning, which is crucial for 3D modeling and generative AI. Therefore, we plan to include relationship understanding into different tasks.

1. **Image generation with the diffusion model**. Diffusion models are powerful in generating high-quality and realistic images by learning data distributions. However, their outputs are often unstructured and lack relational context. Incorporating visual relationships can ensure the generated content adheres to logical or semantic rules, improving their utility in tasks like storytelling, scene synthesis, or AR/VR applications. The integration can be implemented in three ways: (a) Relationship-Aware Conditioning. We can adopt relationship annotations (e.g., Scene Graphs) as part of the conditional input. So to extend text-to-image pipelines to accept structured prompts, like "A dog under a tree; the tree next to a lake". (b) Multi-Modal Embedding Fusion. We can fuse embeddings from visual relationship datasets (like Visual Genome) with learned embeddings from text or vision-language models (e.g., CLIP), which ensures diffusion steps maintain these relational constraints. (c) Loss Functions for Relational Consistency. We can add a loss term during training to penalize outputs that deviate from specified relationships (e.g., an object that violates spatial or functional rules).

2. **3D modeling with Gaussian Splatting**. Gaussian splatting represents 3D scenes using point clouds and Gaussian distributions. It focuses on rendering efficiency and realism in 3D scenes. Embedding visual relationships could improve scene coherence, enabling

relationally aware 3D reconstructions, interactive experiences, or training virtual agents. We can integrate visual relationships into Gaussian Splatting in three ways: (a) Scene Graph Integration. We can use scene graphs to initialize point clouds, with object interactions driving Gaussian placements and densities. For instance, spatial relationships like "on top of," "behind," or "next to" guide point alignment in 3D space. (b) Relational Priors for Splatting. We can incorporate relational priors into splatting by encoding constraints into the neural rendering process. For example, adjust Gaussian parameters (e.g., orientation, scale) to align with expected object relationships. (c) Learning Relationships in 3D. We can train Gaussian splatting models using datasets annotated with 3D spatial relationships.

3. **Object Searching in Complex Scenes**. Understanding perceptual relationships can benefit the robotics field in areas such as object searching, scene understanding, and planning. For instance, knowing the 3D information derived from perceptual relationships enables robots to efficiently locate desired objects. The integration can be implemented in two ways: (a) We can represent environments as graphs where nodes are objects and edges are relationships (e.g., "chair under table"). So we can use these graphs to predict object locations based on relationships observed in prior experiences. (b) Use pre-trained vision-language models (e.g., CLIP) to match object descriptions to observed relationships in the environment.

4. **Medical Imaging**. Perceptual relationships in medical imaging play a vital role in improving the accuracy and efficiency of disease diagnosis. These relationships help analyze not just individual features in an image, but also how these features interact spatially and temporally within anatomical structures. This contextual understanding enhances diagnostic precision, facilitates automated systems, and aids in the early detection of complex diseases. Specifically, we can apply relationships in different tasks: (a) Tumor Detection and Localization. We can use deep learning models to analyze relationships at both the macro (organ-level) and micro (cell-level) scales. Then represent anatomical structures as graphs where nodes represent organs or features and edges capture spatial relationships. With the representations, it can train networks that incorporate surrounding tissue context into tumor classification and segmentation tasks. (b) Diagnosing Lung Diseases. We can embed priors about typical spatial relationships (e.g., "consolidation appears in the

lower lobes in pneumonia"). Then we can use clustering to identify abnormal patterns in the distribution of nodules or infiltrates in lung scans. (c) Neurological Disorders. We can represent the brain's regions as nodes and their functional or structural relationships as edges, then use time-series MRI data to monitor changes in brain structure over time. Therefore, we can detect regional thinning in specific brain areas and their relation to disease progression.

5. **Large multi-modal Foundation Models**. Enhancing relationship understanding can also benefit large multi-modal foundation models. For example, In large vision-language models, accurately identifying relationships improves the model's ability to interpret questions and respond with greater accuracy and supporting evidence. In large language models, understanding relationships ensures that LLMs cite evidence from relevant contexts, improving transparency and trustworthiness.

6. **Fine-grain Relationships in 3D Modeling**. The granularity of relationships, whether coarse or fine-grain, is central to improving 3D understanding. Coarse relationships, such as "the chair is near the table," provide high-level spatial information that defines general layouts, creating a foundation for 3D scene reconstruction. Fine-grain relationships, on the other hand, include intricate details like "the chair is slightly angled toward the table," which refine object placement and add semantic depth to the understanding of interactions within the scene. To utilize these relationships in 3D understanding, we can integrate the parsed scene graphs with geometric data to align spatial relationships with the physical structure of the scene. Coarse relationships provide the general structure for arranging objects, ensuring they are placed in approximately correct locations relative to each other. Fine-grain relationships then enhance the precision of these placements, guiding adjustments in scale, rotation, or orientation to match nuanced constraints. For example, "the vase centered on the table" not only places the vase on the table but also ensures alignment with the table's geometric center.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016.

[2] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. Watch your step: Learning node embeddings via graph attention. In *Advances in Neural Information Processing Systems*, pages 9180–9190, 2018.

[3] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020.

[4] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[5] Somak Aditya, Yezhou Yang, Chitta Baral, Yiannis Aloimonos, and Cornelia Fermüller. Image understanding using vision and reasoning through scene description graph. *Computer Vision and Image Understanding*, 173:33–45, 2018.

[6] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[7] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[8] Martin Andrews, Yew Ken Chia, and Sam Witteveen. Scene graph parsing by attention graph. *arXiv preprint arXiv:1909.06273*, 2019.

[9] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2810, 2018.

[10] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016.

[11] Davide Bacciu, Federico Errica, and Alessio Micheli. Contextual graph markov model: A deep and generative approach to graph processing. In *ICML*, 2018.

[12] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Non-probability sampling network for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[13] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.

[14] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[15] Matthias Bauer, Emilien Dupont, Andy Brock, Dan Rosenbaum, Jonathan Richard Schwarz, and Hyunjik Kim. Spatial functa: Scaling functa to imagenet classification and generation. *arXiv preprint arXiv:2302.03130*, 2023.

[16] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[17] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[18] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012.

[19] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.

[20] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[21] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4217–4229, 2023.

[22] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2028–2038, 2014.

[23] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[24] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10270–10279, 2023.

[25] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.

[26] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.

[27] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2624–2632, 2019.

[28] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019.

[29] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.

[30] Weifeng Chen, Shengyi Qian, and Jia Deng. Learning single-image depth from videos using quality assessment networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5604–5613, 2019.

[31] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[32] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[33] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *Advances in Neural Information Processing Systems*, 35:30923–30936, 2022.

[34] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1791–1800, 2019.

[35] Yu Cheng, Bo Wang, Bo Yang, and Robby T Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1157–1165, 2021.

[36] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021.

[37] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020.

[38] Eugene Choi, Arash Ahmadian, Matthieu Geist, Oilvier Pietquin, and Mohammad Gheshlaghi Azar. Self-improving robust preference optimization. *arXiv preprint arXiv:2406.01660*, 2024.

[39] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[40] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

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

[42] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pages 3076–3086, 2017.

[43] Shengzhe Dai, Li Li, and Zhiheng Li. Modeling vehicle interactions via modified lstm models for trajectory prediction. *IEEE Access*, 7:38287–38296, 2019.

[44] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[45] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[47] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20637–20647, 2023.

[48] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[49] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[50] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[51] Johannes Doellinger, Markus Spies, and Wolfram Burgard. Predicting occupancy distributions of walking humans with convolutional neural networks. *IEEE Robotics and Automation Letters*, 3(3):1522–1528, 2018.

[52] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[53] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[54] Yilun Du, Katie Collins, Josh Tenenbaum, and Vincent Sitzmann. Learning signal-agnostic manifolds of neural fields. *Advances in Neural Information Processing Systems*, 34:8320–8331, 2021.

[55] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *European Conference on Computer Vision*, pages 432–449. Springer, 2025.

[56] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

[57] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.

[58] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Yee Whye Teh, and Arnaud Doucet. Coin++: Data agnostic neural compression. *arXiv preprint arXiv:2201.12904*, 1(2):4, 2022.

[59] Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. Generative models as distributions of functions. *arXiv preprint arXiv:2102.04776*, 2021.

[60] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[61] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[62] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[63] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14300–14310, 2023.

[64] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2020.

[65] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132. IEEE, 2018.

[66] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[67] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH 2011 papers*, pages 1–12. 2011.

[68] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016.

[69] David Forsyth and Jean Ponce. *Computer vision: A modern approach*. Prentice hall, 2011.

[70] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015.

[71] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

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

[73] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2427–2436, 2019.

[74] Huan Fu, Shunming Li, Rongfei Jia, Mingming Gong, Binqiang Zhao, and Dacheng Tao. Hard example generation by texture synthesis for cross-domain shape similarity learning. *Advances in Neural Information Processing Systems*, 33:14675–14687, 2020.

[75] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. Stylegan-human: A data-centric odyssey of human generation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022.

[76] Claudio Gallicchio and Alessio Micheli. Graph echo state networks. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[77] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[78] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1416–1424. ACM, 2018.

[79] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[80] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[81] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning (ICML)*, pages 1243–1252. PMLR, 2017.

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

[83] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[84] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Deep structured implicit functions. *arXiv preprint arXiv:1912.06126*, 2:2, 2019.

[85] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019.

[86] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[87] Shalini Ghosh, Giedrius Burachas, Arijit Ray, and Avi Ziskind. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*, 2019.

[88] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

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

[90] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[91] S Alireza Golestaneh, Saba Dadsetan, and Kris M Kitani. No-reference image quality assessment via transformers, relative ranking, and self-consistency. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3209–3218, 2022.

[92] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[93] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[94] Daniele Grattarola and Pierre Vandergheynst. Generalised implicit neural representations. *Advances in Neural Information Processing Systems*, 35:30446–30458, 2022.

[95] Zongyu Guo, Gergely Flamich, Jiajun He, Zhibo Chen, and José Miguel Hernández-Lobato. Compression with bayesian implicit neural representations. *Advances in Neural Information Processing Systems*, 36:1938–1956, 2023.

[96] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[97] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[98] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[99] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[100] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3279–3286, 2015.

[101] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[102] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[103] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[104] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[105] Stefan Hoermann, Martin Bach, and Klaus Dietmayer. Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2056–2063. IEEE, 2018.

[106] Derek Hoiem, Alexei A Efros, and Martial Hebert. Closing the loop in scene interpretation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[107] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. Sherf: Generalizable human nerf from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9352–9364, October 2023.

[108] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[109] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems*, pages 4558–4567, 2018.

[110] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019.

[111] Zih-Siou Hung, Arun Mallya, and Svetlana Lazebnik. Contextual translation embedding for visual relationship detection and scene graph generation. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3820–3832, 2020.

[112] Seong Jae Hwang, Sathya N Ravi, Zirui Tao, Hyunwoo J Kim, Maxwell D Collins, and Vikas Singh. Tensorize, factorize and regularize: Robust visual relationship learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1023, 2018.

[113] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[114] Junho Jeon and Seungyong Lee. Reconstruction-based pairwise depth dataset for depth image enhancement using cnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 422–438, 2018.

[115] Wei Ji, Yicong Li, Meng Wei, Xindi Shang, Junbin Xiao, Tongwei Ren, and Tat-Seng Chua. Vidvrd 2021: The third grand challenge on video relation detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4779–4783, 2021.

[116] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022.

[117] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[118] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[119] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[120] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[121] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

[122] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6912–6921, 2019.

[123] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[124] Chiheon Kim, Doyup Lee, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Generalizable implicit neural representations via instance pattern composers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11808–11817, June 2023.

[125] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *European conference on computer vision*, pages 143–159. Springer, 2016.

[126] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[127] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[128] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit Cottereau, and Wei Tsang Ooi. Robodepth: Robust out-of-distribution depth estimation under corruptions. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 21298–21342. Curran Associates, Inc., 2023.

[129] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[130] Parth Kothari and Alexandre Alahi. Human trajectory prediction using adversarial loss. In *Proceedings of the 19th Swiss Transport Research Conference, Ascona, Switzerland*, pages 15–17, 2019.

[131] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6867–6876, 2018.

[132] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[133] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[134] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–96, 2014.

[135] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[136] Long Lan, Xinchao Wang, Shiliang Zhang, Dacheng Tao, Wen Gao, and Thomas S. Huang. Interacting tracklets for multi-object tracking. *IEEE Transactions on Image Processing*, 27(9):4585–4597, 2018.

[137] Luca A Lanzendörfer and Roger Wattenhofer. Siamese siren: Audio compression with implicit neural representations. *arXiv preprint arXiv:2306.12957*, 2023.

[138] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4340–4350, January 2023.

[139] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[140] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674. ACM, 2018.

[141] Kuang-Huei Lee, Hamid Palangi, Xi Chen, Houdong Hu, and Jianfeng Gao. Learning visual relation priors for image-text matching and image captioning with neural scene graph generators. *arXiv preprint arXiv:1909.09953*, 2019.

[142] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[143] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[144] Bingyin Li, Xiaoyu Xu, Sheyang Tang, Li Yu, and Zhou Wang. Human perception-guided meta-training for few-shot nerf. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13206–13210. IEEE, 2024.

[145] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1119–1127, 2015.

[146] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11109–11119, 2021.

[147] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[148] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1347–1356, 2017.

[149] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE international conference on computer vision*, pages 1261–1270, 2017.

[150] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[151] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.

[152] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022.

[153] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual relationship detection with deep structural ranking. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[154] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 848–857, 2017.

[155] Julian Lienen, Eyke Hullermeier, Ralph Ewerth, and Nils Nommensen. Monocular depth estimation via listwise ranking using the plackett-luce model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14595–14604, 2021.

[156] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[157] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020.

[158] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1253–1260. IEEE, 2010.

[159] Bo Liu, Qiulei Dong, and Zhanyi Hu. Zero-shot learning from adversarial feature residual to compact visual feature. pages 11547–11554, 2020.

[160] Bo Liu, Qiulei Dong, and Zhanyi Hu. Hardness sampling for self-training based transductive zero-shot learning. pages 16499–16508, 2021.

[161] Bo Liu, Qiulei Dong, and Zhanyi Hu. Semantic-diversity transfer network for generalized zero-shot learning via inner disagreement based ood detector. *Knowledge-Based Systems*, 229:107337, 2021.

[162] Bo Liu, Lihua Hu, Zhanyi Hu, and Qiulei Dong. Hardboost: Boosting zero-shot learning with hard classes. *arXiv preprint arXiv:2201.05479*, 2022.

[163] Huihui Liu, Yiding Yang, and Xinchao Wang. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[164] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[165] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[166] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[167] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6985–6994, 2018.

[168] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023.

[169] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4424–4431, 2019.

[170] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

[171] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

[172] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[173] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[174] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pages 852–869. Springer, 2016.

[175] Yao Lu, Soren Pirk, Jan Dlabal, Anthony Brohan, Ankita Pasad, Zhao Chen, Vincent Casser, Anelia Angelova, and Ariel Gordon. Taskology: Utilizing task relations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8700–8709, 2021.

[176] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022.

[177] Kede Ma, Wentao Liu, Kai Zhang, Zhengfang Duanmu, Zhou Wang, and Wangmeng Zuo. End-to-end blind image quality assessment using deep neural networks. *IEEE Transactions on Image Processing*, 27(3):1202–1213, 2017.

[178] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[179] Andrii Maksai, Xinchao Wang, Francois Fleuret, and Pascal Fua. Non-markovian globally consistent multi-object tracking. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[180] Andrii Maksai, Xinchao Wang, and Pascal Fua. What players do with the ball: A physically constrained interaction modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[181] Michele Mancini, Gabriele Costante, Paolo Valigi, and Thomas A Ciarfuglia. Fast robust monocular depth estimation for obstacle detection with fully convolutional networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4296–4303. IEEE, 2016.

[182] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[183] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European conference on computer vision*, pages 759–776. Springer, 2020.

[184] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7143–7152, 2020.

[185] Armin Masoumian, Hatem A Rashwan, Saddam Abdulwahab, Julian Cristiano, and Domenec Puig. Gcndepth: Self-supervised monocular depth estimation based on graph convolutional network. *arXiv preprint arXiv:2112.06782*, 2021.

[186] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[187] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021.

[188] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2023.

[189] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[190] Alican Mertan, Yusuf Huseyin Sahin, Damien Jade Duff, and Gozde Unal. A new distributional ranking loss with uncertainty: Illustrated in relative depth estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1079–1088. IEEE, 2020.

[191] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019.

[192] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[193] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[194] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

[195] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[196] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020.

[197] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[198] Anush Krishna Moorthy and Alan Conrad Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE transactions on Image Processing*, 20(12):3350–3364, 2011.

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

[200] Quan H Nguyen and William J Beksi. Single image super-resolution via a dual interactive implicit neural network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4936–4945, 2023.

[201] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[202] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[203] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[204] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[205] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. *arXiv preprint arXiv:2204.02296*, 2022.

[206] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[207] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[208] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[209] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[210] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1928–1937, 2017.

[211] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[212] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[213] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2019.

[214] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

[215] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

[216] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection with spatio-temporal graph. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 84–93, 2019.

[217] Jiayan Qiu, Xinchao Wang, Pascal Fua, and Dacheng Tao. Matching seqlets: An unsupervised approach for locality preserving sequence matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[218] Jiayan Qiu, Xinchao Wang, Stephen J Maybank, and Dacheng Tao. World from blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8493–8504, 2019.

[219] Jiayan Qiu, Yiding Yang, Xinchao Wang, and Dacheng Tao. Hallucinating visual instances in total absentia. In *European Conference on Computer Vision*, 2020.

[220] Jiayan Qiu, Yiding Yang, Xinchao Wang, and Dacheng Tao. Scene essence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8322–8333, 2021.

[221] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[222] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[223] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[224] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2349–2359, 2023.

[225] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[226] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019.

[227] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019.

[228] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022.

[229] Eike Rehder, Florian Wirth, Martin Lauer, and Christoph Stiller. Pedestrian prediction by planning using deep neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5903–5908. IEEE, 2018.

[230] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021.

[231] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

[232] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks, 2016.

[233] Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[234] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–771, 2018.

[235] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[236] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[237] Hongning Ruan, Yulin Shao, Qianqian Yang, Liang Zhao, and Dusit Niyato. Point cloud compression with implicit neural representations: A unified framework. *arXiv preprint arXiv:2405.11493*, 2024.

[238] Marc Rußwurm, Konstantin Klemmer, Esther Rolf, Robin Zbinden, and Devis Tuia. Geographic location encoding with spherical harmonics and sinusoidal representation networks. *arXiv preprint arXiv:2310.06743*, 2023.

[239] Mohammad Amin Sadeghi and Ali Farhadi. *Recognition using visual phrases*. IEEE, 2011.

[240] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023.

[241] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[242] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[243] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[244] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[245] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[246] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[247] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015.

[248] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[249] Mohammad Shafiei, Sai Bi, Zhengqin Li, Aidas Liaudanskas, Rodrigo Ortiz-Cayon, and Ravi Ramamoorthi. Learning neural transmittance for efficient rendering of reflectance fields. *arXiv preprint arXiv:2110.13272*, 2021.

[250] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 279–287. ACM, 2019.

[251] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1300–1308, 2017.

[252] Evan Shelhamer, Jonathan T Barron, and Trevor Darrell. Scene intrinsics and depth from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–44, 2015.

[253] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8994–9003, 2021.

[254] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[255] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

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

[257] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[258] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019.

[259] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[260] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[261] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

[262] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[263] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019.

[264] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8:131–162, 2007.

[265] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022.

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

[267] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022.

[268] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3667–3676, 2020.

[269] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34:12278–12291, 2021.

[270] Li Sun, Zhi Yan, Sergi Molina Mellado, Marc Hanheide, and Tom Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5942–5948. IEEE, 2018.

[271] Xu Sun, Tongwei Ren, Yuan Zi, and Gangshan Wu. Video visual relation detection via multi-modal feature fusion. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2657–2661, 2019.

[272] Richard Szeliski. Prediction error as a quality metric for motion and stereo. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 781–788. IEEE, 1999.

[273] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[274] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.

[275] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[276] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[277] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22819–22829, 2023.

[278] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020.

[279] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision*, pages 572–589. Springer, 2022.

[280] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.

[281] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021.

[282] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4190–4200, 2023.

[283] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10424–10433, 2019.

[284] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16773–16783, 2023.

[285] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[286] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[287] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2022–2030, 2018.

[288] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao. Tag disentangled generative adversarial networks for object image re-rendering. In *International joint conference on artificial intelligence (IJCAI)*, 2017.

[289] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2555–2563, 2023.

[290] Peihao Wang, Zhiwen Fan, Tianlong Chen, and Zhangyang Wang. Neural implicit dictionary learning via mixture-of-expert training. In *International Conference on Machine Learning*, pages 22613–22624. PMLR, 2022.

[291] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2809, 2015.

[292] Ruiping Wang, Yong Cui, Xiao Song, Kai Chen, and Hong Fang. Multi-information-based convolutional neural network with attention mechanism for pedestrian trajectory prediction. *Image and Vision Computing*, 107:104110, 2021.

[293] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision and Pattern Recognition (ECCV)*, pages 17–32, 2014.

[294] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2312–2326, 2016.

[295] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[296] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[297] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

[298] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020.

[299] Changhee Won, Jongbin Ryu, and Jongwoo Lim. Omnimvs: End-to-end learning for omnidirectional stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8987–8996, 2019.

[300] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. *Advances in neural information processing systems*, 31, 2018.

[301] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023.

[302] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018.

[303] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 611–620, 2020.

[304] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9421–9431, 2021.

[305] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013.

[306] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[307] Long Xin, Pin Wang, Ching-Yao Chan, Jianyu Chen, Shengbo Eben Li, and Bo Cheng. Intention-aware long horizon trajectory prediction of surrounding ve-

hicles using dual lstm networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1441–1446. IEEE, 2018.

[308] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[309] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[310] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[311] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4479–4489, 2023.

[312] Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, 35:13404–13418, 2022.

[313] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[314] Xiaoyu Xu, Jiayan Qiu, Xinchao Wang, and Zhou Wang. Relationship spatialization for depth estimation. In *European Conference on Computer Vision*, pages 615–637. Springer, 2022.

[315] Hao Yan, Zhihui Ke, Xiaobo Zhou, Tie Qiu, Xidong Shi, and Dadong Jiang. Ds-nerv: Implicit neural video representation with decomposed static and dynamic codes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23019–23029, 2024.

[316] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.

[317] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. 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*, pages 899–908, 2019.

[318] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

[319] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023.

[320] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019.

[321] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[322] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[323] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[324] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. SPAGAN: shortest path graph attention network. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[325] Zuopeng Yang, Daqing Liu, Chaoyue Wang, Jie Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7764–7773, 2022.

[326] Brent Yi, Weijia Zeng, Sam Buchanan, and Yi Ma. Canonical factors for hybrid neural fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3414–3426, 2023.

[327] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision*, pages 263–279. Springer, 2016.

[328] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6796–6807, 2024.

[329] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 322–338, 2018.

[330] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pages 4800–4810, 2018.

[331] Paul Yoo, Jiaxian Guo, Yutaka Matsuo, and Shixiang Shane Gu. Dreamsparse: Escaping from plato's cave with 2d diffusion model given sparse views. *Advances in Neural Information Processing Systems*, 36, 2024.

[332] Tackgeun You, Mijeong Kim, Jungtaek Kim, and Bohyung Han. Generative neural fields by mixtures of neural implicit functions. *Advances in Neural Information Processing Systems*, 36, 2024.

[333] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[334] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, June 2021.

[335] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–87, 2018.

[336] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6490–6499, 2019.

[337] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[338] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *Proceedings of the IEEE international conference on computer vision*, pages 1974–1982, 2017.

[339] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[340] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A

large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023.

[341] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3916–3925, 2022.

[342] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022.

[343] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022.

[344] J. Yue, D. Manocha, and H. Wang. Human trajectory prediction via neural social physics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[345] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015.

[346] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for fast image restoration and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[347] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015.

[348] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering. *arXiv preprint arXiv:1907.12133*, 2019.

[349] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017.

[350] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5678–5686, 2017.

[351] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9185–9194, 2019.

[352] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[353] Kaiwei Zhang, Dandan Zhu, Xiongkuo Min, and Guangtao Zhai. Implicit neural representation learning for hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2022.

[354] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[355] Weixia Zhang, Dingquan Li, Xiongkuo Min, Guangtao Zhai, Guodong Guo, Xiaokang Yang, and Kede Ma. Perceptual attacks of no-reference image quality models with human-in-the-loop. *Advances in Neural Information Processing Systems*, 35:2916–2929, 2022.

[356] Weixia Zhang, Dingquan Li, Guangtao Zhai, Xiaokang Yang, and Kede Ma. Comparison of no-reference image quality models via map estimation in diffusion latents. *arXiv preprint arXiv:2403.06406*, 2024.

[357] Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):36–47, 2018.

[358] Weixia Zhang, Kede Ma, Guangtao Zhai, and Xiaokang Yang. Uncertainty-aware blind image quality assessment in the laboratory and wild. *IEEE Transactions on Image Processing*, 30:3474–3486, 2021.

[359] Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14071–14081, 2023.

[360] Yunfan Zhang, Ties Van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *arXiv preprint arXiv:2112.11312*, 2021.

[361] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018.

[362] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019.

[363] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019.

[364] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12126–12134, 2019.

[365] Wenbo Zhao, Xianming Liu, Zhiwei Zhong, Junjun Jiang, Wei Gao, Ge Li, and Xiangyang Ji. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2007, 2022.

[366] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[367] Sipeng Zheng, Shizhe Chen, and Qin Jin. Vrdformer: End-to-end video visual relation detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18836–18846, 2022.

[368] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

[369] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[370] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[371] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proceedings of the IEEE international conference on computer vision*, pages 589–598, 2017.

[372] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018.

[373] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. arxiv 2019. *arXiv preprint arXiv:1909.08593*, 1909.

[374] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.

[375] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.

[376] Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. Learning ordinal relationships for mid-level vision. In *Proceedings of the IEEE international conference on computer vision*, pages 388–396, 2015.

[377] Jan Zuiderveld, Marco Federici, and Erik J Bekkers. Towards lightweight controllable audio synthesis with conditional implicit neural representations. *arXiv preprint arXiv:2111.08462*, 2021.