applied sciences

MDPI

Article

ED^{2}IF^2-Net: Learning Disentangled Deformed Implicit Fields and Enhanced Displacement Fields from Single Images Using Pyramid Vision Transformer

Xiaoqiang Zhu [1,2], Xinsheng Yao [1], Junjie Zhang [1,*], Mengyao Zhu [1], Lihua You [2], Xiaosong Yang [2], Jianjun Zhang [2], He Zhao [3] and Dan Zeng [1]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; xqzhu@shu.edu.cn (X.Z.); shu_yaoxinsheng@shu.edu.cn (X.Y.); zhumengyao@shu.edu.cn (M.Z.); dzeng@shu.edu.cn (D.Z.)
[2] National Center for Computer Animation, Bournemouth University, Bournemouth BH12 5BB, UK; lyou@bournemouth.ac.uk (L.Y.); xyang@bournemouth.ac.uk (X.Y.); jzhang@bournemouth.ac.uk (J.Z.)
[3] R&D Department, Changzhou Micro-Intelligence Co., Ltd., Changzhou 213002, China; he.zhao@micro-i.com.cn
* Correspondence: junjie_zhang@shu.edu.cn

Abstract: There has emerged substantial research in addressing single-view 3D reconstruction and the majority of the state-of-the-art implicit methods employ CNNs as the backbone network. On the other hand, transformers have shown remarkable performance in many vision tasks. However, it is still unknown whether transformers are suitable for single-view implicit 3D reconstruction. In this paper, we propose the first end-to-end single-view 3D reconstruction network based on the Pyramid Vision Transformer (PVT), called ED^{2}IF^2-Net, which disentangles the reconstruction of an implicit field into the reconstruction of topological structures and the recovery of surface details to achieve high-fidelity shape reconstruction. ED^{2}IF^2-Net uses a Pyramid Vision Transformer encoder to extract multi-scale hierarchical local features and a global vector of the input single image, which are fed into three separate decoders. A coarse shape decoder reconstructs a coarse implicit field based on the global vector, a deformation decoder iteratively refines the coarse implicit field using the pixel-aligned local features to obtain a deformed implicit field through multiple implicit field deformation blocks (IFDBs), and a surface detail decoder predicts an enhanced displacement field using the local features with hybrid attention modules (HAMs). The final output is a fusion of the deformed implicit field and the enhanced displacement field, with four loss terms applied to reconstruct the coarse implicit field, structure details through a novel deformation loss, overall shape after fusion, and surface details via a Laplacian loss. The quantitative results obtained from the ShapeNet dataset validate the exceptional performance of ED^{2}IF^2-Net. Notably, ED^{2}IF^2-Net-L stands out as the top-performing variant, exhibiting the highest mean IoU, CD, EMD, ECD-3D, and ECD-2D scores, reaching impressive values of 61.1, 7.26, 2.51, 6.08, and 1.84, respectively. The extensive experimental evaluations consistently demonstrate the state-of-the-art capabilities of ED^{2}IF^2-Net in terms of reconstructing topological structures and recovering surface details, all while maintaining competitive inference time.

Keywords: 3D reconstruction; single-view; deep learning; computer vision; transformer; implicit field; signed distance function; displacement field

Citation: Zhu, X.; Yao, X.; Zhang, J.; Zhu, M.; You, L.; Yang, X.; Zhang, J.; Zhao, H.; Zeng, D. ED^{2}IF^2-Net: Learning Disentangled Deformed Implicit Fields and Enhanced Displacement Fields from Single Images Using Pyramid Vision Transformer. *Appl. Sci.* **2023**, *13*, 7577. https://doi.org/10.3390/app13137577

Academic Editor: Antonio Fernández-Caballero

Received: 25 May 2023
Revised: 16 June 2023
Accepted: 23 June 2023
Published: 27 June 2023

1. Introduction

Single-view 3D reconstruction aims to reconstruct object shapes from single-view RGB images, which are widely used in areas such as robotics, VR, and autonomous driving. However, single-view 3D reconstruction faces challenges that stem from its intrinsic ill-posed nature. A large number of single-view 3D reconstruction approaches have been presented

recently. Among them, deep-learning-based methods [1–35] have yielded more advanced results. Shape representations in deep-learning-based approaches can be classified into explicit [1–17] and implicit [18–35] representations, and the latter ones are independent of memories and resolutions, which can significantly improve the reconstruction performance.

The main ideas of the earlier data-driven implicit approaches [18–22] are to learn latent vectors of the input images and the neural networks are applied to fit the mapping relationship from the query points to a implicit scalar field. For example, DeepSDF [18] introduces latent codes that are able to represent similar objects and output signed distance functions (SDFs) approximating object shapes in combination with query point coordinates. IM-Net [19] encodes a single input image to extract a latent vector, which is then decoded together with the query point coordinates to generate an implicit scalar field value representing the spatial relationship between the point and the object shape. Occupancy Network [20] encodes different types of inputs into embeddings while converting query points into point features, and the decoder incorporates all the information and outputs a real number to indicate the occupancy probability of the query point. Littwin et al. [21] use the encoding vector of the input image as the weight matrix in an MLP for binary classification of query points, resulting in the generation of an implicit field. These methods can only reconstruct the coarse shape but fail to reproduce the details of the object. Rather than predicting a single global implicit field, PQ-NET [23] outputs a SDF for each intrinsic structure of the object and fuses these implicit fields to generate the final SDF, producing more promising reconstruction results.

Recently, several novel CNN-based models have been proposed [24–29]. DISN [24] fuses global and local features of the input image with point features of query points to obtain the fused SDF. MDISN [25] deforms a randomly generated SDF based on local feature variations at the layer level to approximate the ground-truth SDF. Ladybird [26] considers pixel-aligned local features of query points and their symmetry points, combining them with global features to output the SDF. Ray-ONet [27] integrates global features, local features, and scaling parameters to estimate the occupancy probability of spatial query points along rays, reducing complexity and improving performance compared to Occupancy Network [20]. Peng et al. [28] merge global and local features extracted by an encoder, incorporate query points via linear interpolation, and use subsequent networks to predict occupancy values. In contrast to earlier works [18–23], such methods [24–28] can better capture structure details and recover finer shapes owing to the integration of local features. However, details at the surface level such as depth, which are equally critical for visual perception, are still poorly reconstructed. D^2IM-Net [29] focuses on recovering surface details, which often produces promising surface features, yet is unable to reconstruct the correct topological structures. Furthermore, previous CNN-based approaches [24–29] often encounter two inherent limitations associated with the convolutional layer. Firstly, convolutional kernels treat all pixels equally, resulting in inefficiency when processing images. This uniform treatment fails to capture the varying importance and dependencies of different pixels within an image. Secondly, due to the local nature of convolution, long-range pixel relationships are not effectively modeled. As a result, crucial contextual information may be overlooked, hindering the ability to fully understand and exploit the complex dependencies and interactions between pixels across the entire image. The PIFu series [30–32] incorporates pixel-aligned local features and depth information into their paradigm, with a primary focus on human reconstruction.

With the advent of the works like ViT [36] and DeiT [37], transformers have obtained considerable attention in computer vision recently. Transformer-based vision models have achieved state-of-the-art performance in several downstream tasks, such as DETR [38] for object detection, SwinIR [39] for image restoration, Segmenter [40] for semantic segmentation, and MViTv2 [41] for image classification. While transformers have demonstrated preliminary success in many tasks including explicit 3D reconstruction [1–5,42], whether they could be successfully employed to improve implicit 3D reconstruction is still unknown.

To address the limitations of existing implicit methods that struggle to simultaneously reconstruct the topological structure and surface details of objects, ED^{2}IF^2-Net is proposed in this paper. Our approach utilizes transformers, specifically Pyramid Vision Transformer (PVT), to enable end-to-end single-view implicit 3D reconstruction. By leveraging PVT, we aim to mitigate the negative impacts of underlying convolutional layers in CNN-based methods, allowing for comprehensive reconstruction of both topological structures and surface details from a single image. For an input image, local features and a global vector are extracted using a pre-trained Pyramid Vision Transformer encoder [43]. Subsequently, a coarse shape decoder reconstructs a coarse implicit field based on the global vector. A deformation decoder, incorporating symmetry priors that provide extra knowledge about the object shape, predicts a deformed implicit field with finer-grained structure details using pixel-aligned local features and multiple implicit field deformation blocks (IFDBs). Finally, a surface detail decoder equipped with hybrid attention modules [44] (HAMs) constructs an enhanced displacement field, enabling the recovery of enhanced surface details from the local features. In order to facilitate the learning of the implicit field deformation function, IFDB offers a lightweight and effective approach. It refines the coarse implicit field by leveraging information from query points and pixel-aligned local features at neighboring scales. Through simple iterations, multiple IFDBs efficiently fit the deformation function, enabling the generation of the deformed implicit field that captures the finer topological structure of the object. In contrast to CBAM [45], HAM is a more novel and parameter-efficient module that significantly improves surface detail recovery performance. The output of the proposed ED^{2}IF^2-Net is a fusion of the deformed implicit field and the enhanced displacement field together. The main contributions of this paper include:

1. A Pyramid-Vision-Transformer-based ED^{2}IF^2-Net is proposed for end-to-end single-view implicit 3D reconstruction, which disentangles implicit field reconstruction into accurate topological structures and enhanced surface details with competitive inference time. To our knowledge, it is the first method to utilize transformers for single-view implicit 3D reconstruction. Experimental results show superior performance in both overall reconstruction and detail recovery.
2. The finer topological structural details of the object are achieved through iterative refinement of the coarse implicit field using multiple IFDBs. IFDB deforms the implicit field from coarse to fine based on query point and pixel-aligned local feature variations at continuous scales. ED^{2}IF^2-Net also enhances surface detail representation at spatial and channel levels.
3. A novel loss function consisting of four terms is proposed, where coarse shape loss and overall shape loss allow the reconstruction of the coarse shape and the overall shape after fusion, and novel deformation loss and Laplacian loss enable ED^{2}IF^2-Net to reconstruct structure details and recover surface details, respectively.

2. Related Works

Since the proposed ED^{2}IF^2-Net is a single-view 3D reconstruction network based on the Pyramid Vision Transformer, in this section, we review some related works as follows.

2.1. Shape Representations

The shape of an object can be represented by voxels [1–10], point clouds [11–14], meshes [15–17], and implicit functions [18–35]. In this paper, with respect to shape representations, we choose implicit functions, as they can process arbitrary topologies and support multi-resolution representations in comparison to others.

2.2. Implicit Methods for Single-View 3D Reconstruction

Since the object shapes in this work are represented with implicit functions, this section focuses on reviewing the implicit methods for single-view 3D reconstruction.

There are mainly two popular forms of data in deep-learning-based single-view implicit 3D reconstructions: occupancy probability and SDF. Specifically, the implicit models learn the scalar value of each query point under the supervision of the ground-truth occupancy probability or SDF. Earlier implicit methods, such as Occupancy Network [20] and IM-Net [19], tend to adopt a straightforward idea. There, the latent vectors of the input image are firstly extracted via an image encoder, and are subsequently combined with the features or coordinates of the query points for the MLP inputs, and then the occupancy probability or SDF for each query point can be predicted. Recently, a few novel CNN-based implicit 3D reconstruction models [24–29] have been proposed that take into account local features, resulting in more promising reconstruction performance.

The most relevant works to ours are DISN [24], MDISN [25], and D^{2}IM-Net [29]. Specifically, both DISN and MDISN predict the camera parameters of the input image to extract local features corresponding to each query point. In DISN, query point features are concatenated with global and local features. Two concatenated features are decoded to obtain two predicted values, which are summed to derive the final SDF. MDISN deforms the randomly generated SDF for each query point from coarse resolutions to fine ones depending on the variation of local features. D^{2}IM-Net predicts the camera pose and decomposes the reconstruction of the object's implicit field into two parts: the reconstruction of coarse shapes and the recovery of details. DISN and MDISN achieve better experimental results than earlier approaches, yielding shapes with more structure details. However, they still fail to recover the surface details of an object. While D^{2}IM-Net is capable of recovering good surface details, it often results in poor topological structures. Although a variant of D^{2}IM-Net, called D^{2}IM-Net$_{GL}$, uses both global and local features in the basic decoder, it still struggles to reconstruct a satisfactory shape and even produces blurry surface details.

Compared to DISN, MDISN and D^{2}IM-Net, ED^{2}IF^2-Net is the first to employ a transformer to solve single-view implicit 3D reconstruction, which alleviates the negative effects brought by convolution in CNN-based models. This paper proposes a novel paradigm that disentangles the reconstruction of an object into reconstruction of more accurate topological structures and enhanced surface details. The finer-grained topological structure details and enhanced surface details are obtained through iterative refinement of the coarse implicit field using the multiple IFDBs, as well as enhancement of the surface detail features in both spatial and channel dimensions using HAMs, as shown in Figure 1. The core difference lies in the construction of individual specific loss terms for all learned fields, including the coarse implicit field, deformed implicit field, and enhanced displacement field. This disentanglement of the deformed implicit field, which contains most of the topological structures, from the enhanced displacement field allows for better learning, resulting in the recovery of enhanced surface details. Actually, a novel deformation loss for learning structure details from ground truth is introduced in our combined loss function, while the surface details can be learned from ground-truth normal maps by applying a Laplacian loss. Extensive qualitative and quantitative comparisons conducted in the experimental Section 4 unequivocally demonstrate the remarkable capabilities of our proposed ED^{2}IF^2-Net. In stark contrast to the limitations observed in DISN and MDISN, where surface detail recovery of objects is lacking, ED^{2}IF^2-Net successfully overcomes this challenge. The reconstructed results exhibit significantly improved surface detail fidelity, showcasing the effectiveness and superiority of our approach. Furthermore, our model solves the problem of D^{2}IM-Net and its variant D^{2}IM-Net$_{GL}$, which reconstruct the wrong topological structures. ED^{2}IF^2-Net is capable of generating visually attractive and high-quality 3D shapes.

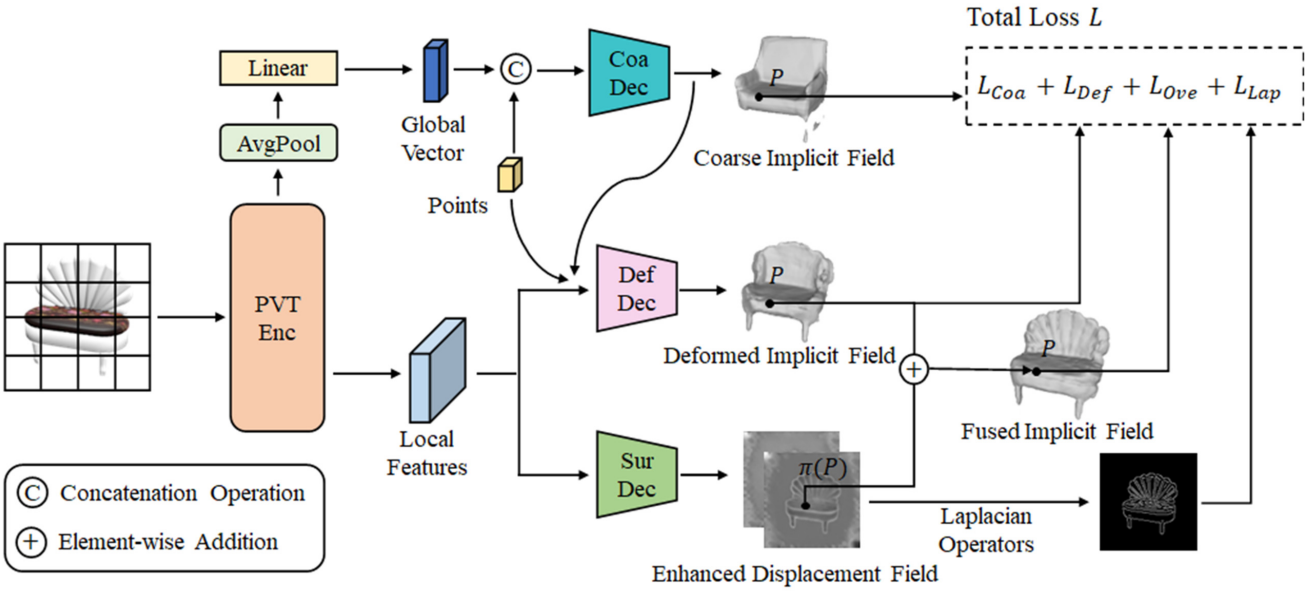

Figure 1. The overall pipeline of the proposed ED^{2}IF^2-Net, where P is a 3D query point and $\pi(\cdot)$ represents the operation of projecting a 3D spatial query point to an image. PVT Enc means Pyramid Vision Transformer encoder. Coa Dec, Def Dec, and Sur Dec denote Coarse Shape Decoder, Deformation Decoder, and Surface Detail Decoder, respectively. ED^{2}IF^2-Net first extracts a global vector and the local features of the input image via a Pyramid Vision Transformer encoder. The global vector is used in a coarse shape decoder to predict a coarse implicit field, which is then iteratively refined by a deformation decoder to obtain a deformed implicit field with finer structure details using multiple implicit field deformation blocks (IFDBs). A surface detail decoder with hybrid attention modules (HAMs) uses local features to recover an enhanced displacement field. The final output of ED^{2}IF^2-Net is a fusion of the deformed implicit field and the enhanced displacement field. Four combined loss terms are applied to reconstruct the coarse implicit field, structure details, overall shape, and surface details.

2.3. Laplacian Operators

Laplacian operators are frequently used to extract local variations in images and 3D shapes. Further, Laplacian pyramids have so far been used extensively in neural models for super-resolution image reconstruction [46,47] and generation [48] by extracting multi-scale structures from images. Li et al. [49] propose a Laplacian loss for image synthesis, which effectively preserves image details and eliminates artifacts. Recently, there have been some works on single-view 3D reconstruction using Laplacian operators. Wang et al. [17] apply Laplacian loss to meshes by minimizing the loss between Laplacian coordinates before and after surface mesh deformation. Liu et al. [50] smooth the surface via a Laplacian regularization, but it is prone to lose the surface details of the object. D^{2}IM-Net [29] takes the disentangled detail information as a displacement field, which recovers the surface details well with Laplacian loss. In this work, we follow and improve D^{2}IM-Net regarding Laplacian loss. The key difference is that, in our acquisition of the displacement field, the feature representation of surface details is enhanced using HAM [44], which effectively overcomes the lack of surface details in D^{2}IM-Net.

2.4. Transformers in Computer Vision

Transformers [51] originate from natural language processing whose core component is multi-head self-attention. Recently, transformers have received much attention in computer vision and have made a profound impact. For a comprehensive review of transformers in vision, the readers are referred to [52]. For applications in vision, transformers have achieved state-of-the-art performance in object detection [38], image classification [36], image restoration [39], and multi-view 3D reconstruction [2]. In this work, a Pyramid

Vision Transformer [43] is used to extract multi-scale hierarchical local features and a global vector. PVT inherits the advantages of CNNs and transformers in that it can extract multi-scale hierarchical local features from images without inductive bias. Ablation studies also demonstrate that, when the Pyramid Vision Transformer is used as an encoder for ED^{2}IF^2-Net, fewer artifacts and better performance can be achieved compared to ResNet18 [53].

3. Methodology

3.1. Overview

In this work, we aim at reconstructing high-fidelity 3D shapes with topological structures and surface details by means of a network that models the signed distance function (SDF) defined as g, given a single RGB image $I \in \mathbb{R}^{H\times W\times 3}$ of the object and any spatial query point $P \in \mathbb{R}^3$. The network outputs the signed distance function values $s = g(I, P), s \in \mathbb{R}$. The training data pair for ED^{2}IF^2-Net to learn the implicit function is made up of single-view images of the object, spatial query points, and their corresponding ground-truth SDF values, viz. $(I, P, SDF(P))$.

ED^{2}IF^2-Net disentangles the SDF of the shape T into the deformed implicit field with structure details and the enhanced displacement field that allows the surface details of the object to be reestablished. The pipeline of ED^{2}IF^2-Net is shown in Figure 1. ED^{2}IF^2-Net extracts image features with a Pyramid Vision Transformer encoder followed by three decoders reconstructing the coarse implicit field, the deformed implicit field, and the enhanced displacement field, respectively. Then the latter two scalar fields are fused to get the final SDF. Finally, the iso-surface with $SDF = 0$ can be extracted using Marching Cubes [54] for visualization.

The following sections describe in detail how ED^{2}IF^2-Net disentangles the implicit field, network architecture, and loss function.

3.2. Disentanglement Method

The variations of the detailed information around the surface of the object (i.e., surface details) affect the Laplacian of the SDF [55]. Inspired by this, the surface details of an object can be detected through Laplacian operators and the remaining topological structures can be reconstructed according to an appropriate loss function, thus disentangling the implicit field reconstruction into topological structure reconstruction and surface detail recovery. As shown in Figure 2, the ground-truth SDF is disentangled into the deformed implicit field with structure details and the enhanced displacement field containing surface details of the object. The most similar work to ours is D^{2}IM-Net [29], which only disentangles the ground-truth SDF into the sum of a coarse implicit field and a displacement field. Unlike D^{2}IM-Net, our disentangled deformed implicit field is based on the coarse implicit field, where the deformed implicit field contains most of the topological structures. Given a query point P, our disentanglement solution can be denoted as:

$$SDF(P) = g_{su}(P) + g_{st}(P), \tag{1}$$

$$f : g_{co} \in \mathbb{R} \mapsto g_{st} \in \mathbb{R}, \tag{2}$$

where SDF denotes the ground-truth SDF, g_{su}, g_{st}, and g_{co} represent the enhanced displacement field with surface details, the deformed implicit field containing most of topological structures, and the coarse implicit field, respectively, and f defines the deformation function from g_{co} to g_{st}.

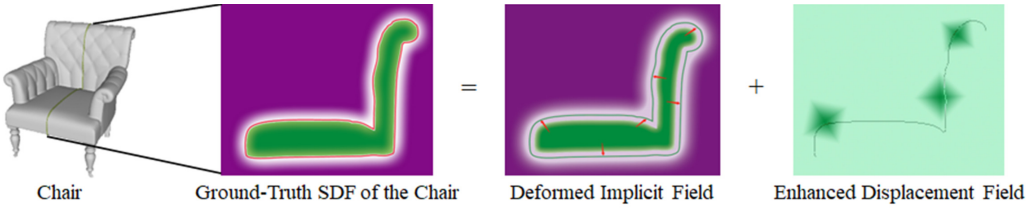

Figure 2. An illustrative description of our disentanglement. ED^{2}IF^2-Net disentangles the ground-truth SDF of the chair into a deformed implicit field and an enhanced displacement field (visible surface), where the deformed implicit field is obtained by refining the coarse implicit field of the object. The red arrows in the deformed implicit field represent the deformation function f from the coarse implicit field (green part) to the deformed implicit field (containing most of the topological structures of the object).

Actually, we can suppose that the shape embedded in the deformed implicit field is smooth and the reconstructed shape from the deformed implicit field can only approximate the object surface. Therefore, the surface details can be further represented with the enhanced displacement field. As the enhanced displacement field is attached onto the smooth deformed implicit field near the iso-surface of the object, the Laplacian of the enhanced displacement field is approximately equal to the Laplacian of SDF:

$$\triangle SDF(P) = \triangle g_{su}(P). \tag{3}$$

In order to accelerate the network training, Laplacian for only the sampling points, whose minimum distance to the object shape T is less than a predefined threshold α, will be taken into consideration.

Motivated by the works [29,31,56] related to inference on the visible and invisible surfaces of objects, the forward and backward displacement maps are introduced for the visible and occluded parts of the object, respectively. Our forward displacement map recovers the visible surface details of the object based on a Laplacian loss and the backward displacement map is used to fine-tune the deformed implicit field, further compensating for unreconstructed structure details and fixing incorrect topological structures. In short, we have

$$SDF(P) = \begin{cases} g_{st}(P) + g_{suF}(p), if P \in P_V, \\ g_{st}(P) + g_{suB}(p), otherwise, \end{cases} \tag{4}$$

$$\triangle g_{suF}(p) = \triangle SDF(P), P \in P_V, \tag{5}$$

where g_{suF} and g_{suB} represent the forward and backward displacement maps, respectively. $p = \pi(P)$ is the projection of P on the single-view image. P_V is the point set which consists of points close to the visible surface of the object.

Indeed, 3D displacement fields are more direct and are also defined in 3D space. However, displacement maps are applied instead of 3D displacement fields because they enable alignment of the input image with the details, making it possible to calculate the Laplacian loss term. Additionally, it is more intuitive for us to observe the detailed information of the object in the displacement maps.

3.3. Network Architecture

The proposed ED^{2}IF^2-Net contains four main components: Pyramid Vision Transformer encoder, coarse shape decoder, deformation decoder, and surface detail decoder.

Based on the Pyramid Vision Transformer encoder, two variants of ED^{2}IF^2-Net are designed: ED^{2}IF^2-Net-T with lower computation complexity and ED^{2}IF^2-Net-L with higher computation complexity, and the latter achieves more pleasing reconstruction results. ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L differ only in encoders, and they share the same architecture for the other parts.

3.3.1. Pyramid Vision Transformer Encoder

In this work, a Pyramid Vision Transformer [43] is used as an encoder for image feature extraction, which consists of four stages. Each of these stages is composed of a patch embedding layer and transformer encoder layers extracting multi-scale local features. In the k^{th} stage, the patch embedding layer partitions the input $M_{k-1} \in \mathbb{R}^{H_{k-1} \times W_{k-1} \times D_{k-1}}$ into a total of $\frac{H_{k-1}W_{k-1}}{B_k^2}$ patches, assuming that the size of each patch after partition is B_k. Then, these patches are flattened, followed by a linear projection to the corresponding dimension D_k of the current stage. After that, embedded patches are reshaped to $\frac{H_{k-1}}{B_k} \times \frac{W_{k-1}}{B_k} \times D_k$, where the width and the height are scaled by a factor of B_k, and later fed into transformer encoder layers together with the position embeddings. In this way, the image features with different scales can be generated at different stages.

In addition, one of the core components of transformer encoder layers in the Pyramid Vision Transformer [43] is Spatial-Reduction Attention (SRA) that can extract high-resolution features without too much computation complexity. The input of SRA is a query vector Q, a key vector K, and a value vector V. It differs from the standard MSA only in that an extra spatial reduction is performed on K and V before the standard multi-head self-attention. The spatial reduction can be described as:

$$SR(X) = LN(Reshape(X, R_k)W_L), \tag{6}$$

where $X \in \mathbb{R}^{(H_k W_k) \times D_k}$ indicates the input to be reduced, R_k is a hyperparameter that represents the reduction factor of the k^{th} stage, LN denotes layer normalization, $Reshape(X, R_k)$ means the operation of transforming the input X into a sequence $S \in \mathbb{R}^{(\frac{H_k W_k}{R_k^2}) \times (R_k^2 D_k)}$, and $W_L \in \mathbb{R}^{(R_k^2 D_k) \times D_k}$ represents the linear projection function for changing the size of S to $(\frac{H_k W_k}{R_k^2}) \times D_k$.

In our implementation, two pre-trained models of the Pyramid Vision Transformer [43] are used, PVT-Tiny and PVT-Large, as encoders for ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L, respectively. The dimensions of local features for all stages of PVT-Tiny and PVT-Large are 64, 128, 320, and 512. The Pyramid Vision Transformer encoder finally outputs local features at four scales denoted as $l_i (i \in \{0, 1, 2, 3\})$ and a global vector z of the input single-view image.

3.3.2. Coarse Shape Decoder

Inspired by IM-Net [19], implicit 3D reconstruction is by nature a classification problem and we use $ReLU$ for nonlinear activation of the $MLPs$ to fit the SDF of the object's coarse shape. Specifically, the global vector z and the query point P are concatenated together as the input, and the coarse implicit field g_{co} will be output through the $MLPs$:

$$g_{co}(P) = MLPs(concat(z, P)). \tag{7}$$

However, the coarse implicit field g_{co} is merely capable of approximating the coarse shape of the object, and unable to reconstruct the structure details and recover the surface details. Therefore, a deformation decoder and a surface detail decoder can be applied to learn details of structure and surface of the object, respectively.

3.3.3. Deformation Decoder

The deformation function f can be learnt via the deformation decoder, as illustrated in Figure 3. The deformation decoder firstly unifies the multi-scale local features through a bilinear interpolation and then retrieves the local features for the query point P at all scales in a pixel-aligned manner. Let $p = \pi(P)$ be the projection of P on the image. Following the similar idea of Ladybird [26], two pixel-aligned local features h_1 and h_2 are provided for P. Specifically, h_1 and h_2 are extracted from the projection of P and its self-reflecting symmetric point P_s on l_i, respectively, which are then concatenated as the final pixel-aligned local feature $l_i(p)$ of P. Finally, the continuous pixel-aligned local feature pairs are used

to refine the coarse implicit field g_{co} through the core lightweight components of the deformation decoder, i.e., implicit field deformation block (IFDB). There exist three IFDBs in our implementation of the deformation decoder and the last one generates the deformed implicit field g_{st} (see Algorithm 1):

$$g_{st}(P) = f(g_{co}(P), l_0, l_1, l_2, l_3, P, p), \tag{8}$$

$$s_1(P), c_1 = IFDB(g_{co}(P), l_2(p), l_3(p), P), \tag{9}$$

$$s_2(P), c_2 = IFDB(s_1(P), l_1(p), l_2(p), P, c_1), \tag{10}$$

$$g_{st}(P), c_3 = IFDB(s_2(P), l_0(p), l_1(p), P, c_2), \tag{11}$$

where $s_j(P)$ and c_j ($j \in \{1, 2, 3\}$) stand for the intermediate implicit field at P and the state code generated by the j^{th} IFDB, respectively, in particular, $g_{st}(P) = s_3(P)$.

Algorithm 1 Deformation

Input: coarse implicit field g_{co}, multi-scale local features $l_i (i \in \{0, 1, 2, 3\})$, query point P and its projection p on the image
Output: deformed implicit field g_{st}

function DEFORMATION(g_{co}, l_i, P, p)
  $P_s \leftarrow Find\_Symmetry\_Point(P)$ // Find the symmetry point of the query point P
  $p_s \leftarrow Find\_Symmetry\_Point\_Projection(P_s)$ // Find the projection of the symmetry point of the query point P on the image
  for $i = 0 \rightarrow 3$ **do**
    $res \leftarrow 224$
    $l_i \leftarrow Bilinear\_Interpolation(l_i, res)$ // Unification of multi-scale local features to res through bilinear interpolation
    $h_1 \leftarrow Grid\_Sampling(l_i, p)$
    $h_2 \leftarrow Grid\_Sampling(l_i, p_s)$
    $l_i(p) \leftarrow Concatenate(h_1, h_2)$
  end for
  for $j = 1 \rightarrow 3$ **do**
    if j == 1 **then**
      $s_0(P) = g_{co}(P)$
      $c_0 = 0$
    end if
    $s_j(P), c_j \leftarrow IFDB(s_{j-1}(P), l_{3-j}(p), l_{4-j}(p), P, c_{j-1})$
    if j == 3 **then**
      $g_{st}(P) \leftarrow s_j(P)$
    end if
  end for
  return $g_{st}(P)$
end function

Pixel-aligned local feature pairs not only enable the deformed implicit field to reconstruct the finer-grained topological structure details aligned with the image, but also guarantee that the surface details can be correctly recovered using the surface detail decoder. This is achieved by incorporating additional information about the query point and its symmetry point in the object shape into the features.

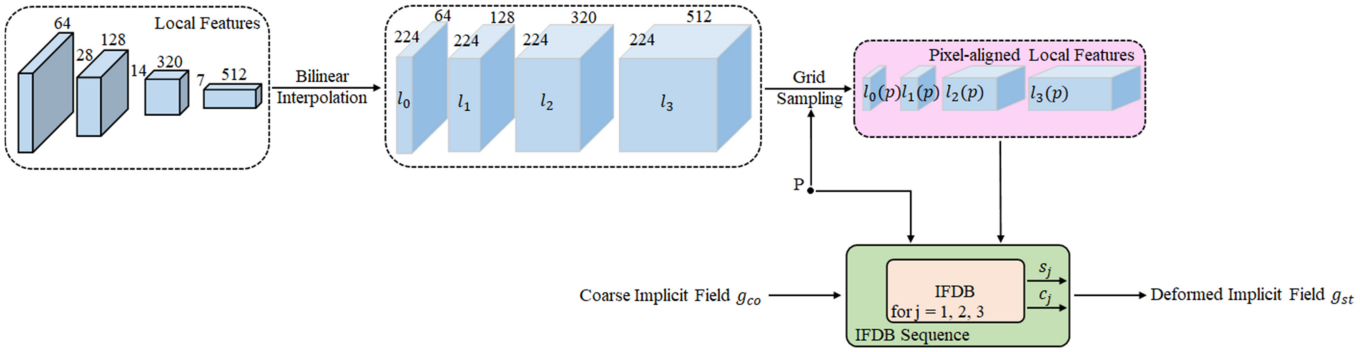

Figure 3. Architecture of the deformation decoder, where s_j and c_j represent the intermediate implicit field and the state code of the j^{th} IFDB output, respectively.

Implicit Field Deformation Block

It is known that the local features with larger scale tend to produce an overall shape, while the ones with smaller scale can keep fine-grained structure details. The IFDB takes advantage of this characteristic, which can be seen in Figure 4. To ensure a smooth implicit field deformation, IFDB deforms the input implicit field according to the variations of the pixel-aligned local features between adjacent scales. Moreover, a state code is used to record all the information of the current implicit field deformation, which will be updated at the end of each IFDB for the next IFDB. In contrast, the coordinates of the query points are input into the deformation module instead of inputting point features in MDISN [25], and our policy performs better than the latter.

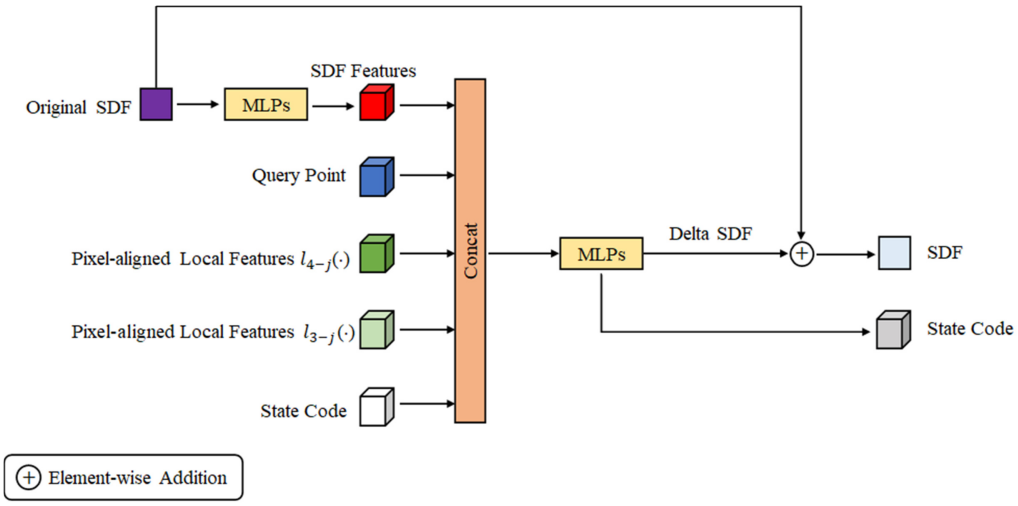

Figure 4. Illustrations of the j^{th} IFDB, where Concat means concatenation operation.

3.3.4. Surface Detail Decoder

In order to recover the enhanced displacement field g_{su} with surface details, a surface detail decoder is applied to recover the detailed displacement maps of an object. As demonstrated in Figure 5, a surface detail decoder takes as input all local features that are extracted by the hybrid attention module [44] (HAM) consisting of spatial and channel attention to enhance the feature representation of surface details. Then, a 1×1 convolution layer and a *ReLU* activation layer are employed to decrease the channels, followed by an upsampling and a 1×1 convolution layer. After that, the outputs of the convolution are element-wise accumulated into the features of the next scale. After repeating the above workflow three times, the features are upsampled twice to keep the consistent size with the input image, followed by a series of convolution and HAM layers. Finally, the enhanced forward and backward displacement maps are output through a 1×1 convolution layer.

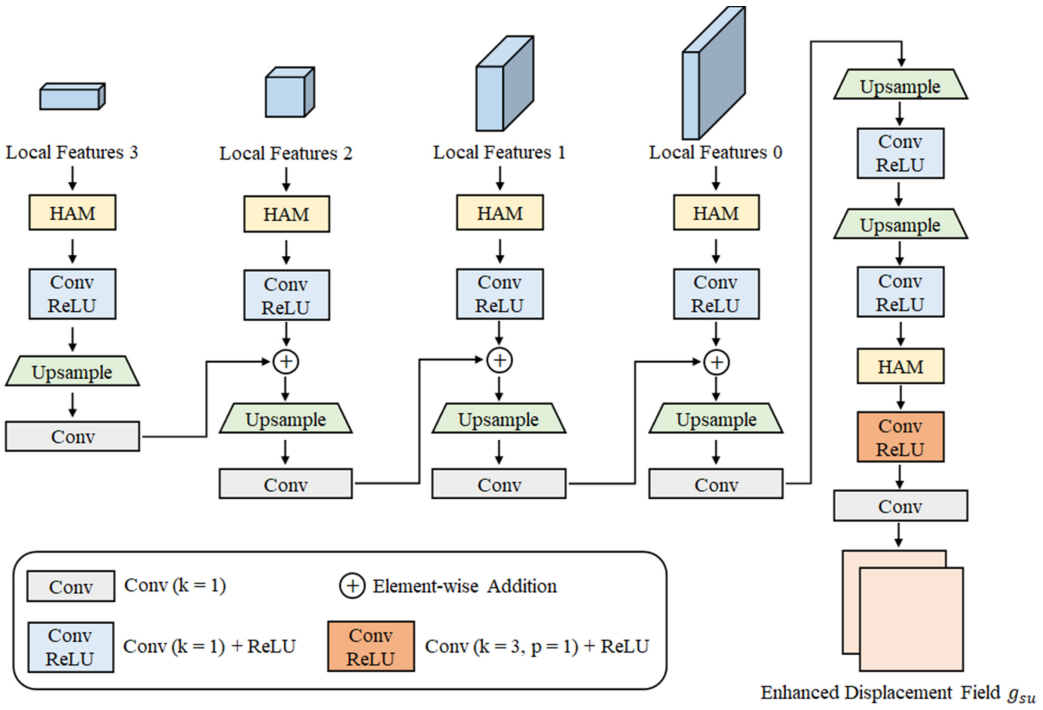

Figure 5. Architecture of the surface detail decoder.

According to Equation (4), if P is near the visible surface, the deformed implicit field of P is added to the forward displacement map at p. Conversely, it is added to the backward one.

In our implementation, the gradient of the SDF on each query point is derived using a central difference approximation. In the case that the direction of the gradient is approaching the viewpoint orientation and the ground-truth SDF is less than a specific threshold, the point is considered as being close to the visible surface. Otherwise, the point is treated as being near the invisible surface. Moreover, a similar network to DISN [24] for estimating the camera parameters is also trained. It should be pointed out that the camera parameters and the gradients derived from the ground-truth SDF used in training are the ground truth, and the predicted values are used in testing.

3.4. Loss Function and Sampling Strategy

The total loss function of ED^{2}IF^2-Net consists of four components $L = L_{Coa} + L_{Def} + L_{Ove} + L_{Lap}$, where L_{Coa}, L_{Def}, L_{Ove}, and L_{Lap} represent the coarse shape loss, deformation loss, overall shape loss, and Laplacian loss, respectively. More specifically, L_2-norm-based L_{Coa} is used to minimize the distance between the coarse implicit field g_{co} and the ground-truth SDF SDF, and L_1-norm-based L_{Ove} is employed to minimize the distance between the fused implicit field g and SDF, which can regularize the enhanced displacement field:

$$L_{Coa} = \frac{1}{N}\sum_{i=1}^{N} ||g_{co}(P_i) - SDF(P_i)||_2^2, \tag{12}$$

$$L_{Ove} = \frac{1}{N}\sum_{i=1}^{N} |g(P_i) - SDF(P_i)|. \tag{13}$$

The structure details are evaluated through a novel deformation loss L_{Def}. Since the deformation decoder iteratively refines g_{co}, the intermediate implicit field s generated by

all IFDBs, and the deformed implicit field g_{st} are all taken into consideration, and their L_1-distances to SDF are accumulated through a weighted summation:

$$L_{Def} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\omega_j|s_j(P_i) - SDF(P_i)| + \frac{1}{N}\sum_{i=1}^{N}\omega_0|g_{st}(P_i) - SDF(P_i)|, \tag{14}$$

$$\sum_{j=0}^{M}\omega_j = 1, \tag{15}$$

where P_i represents the i^{th} query point, N denotes the number of query points, M indicates the number of intermediate implicit fields, the j^{th} intermediate implicit field is defined as s_j, ω_j stands for the weight assigned to s_j in the deformation loss, specifically, and ω_0 is the weight of the deformed implicit field g_{st}.

L_{Lap} is the L_2-distance between the Laplacian of the forward displacement map g_{suF} and the Laplacian of SDF. As these two Laplacians are not in the same space, the forward displacement map g_{suF} is a 2D image, whereas SDF is in 3D space. This problem is addressed using the Laplacian of SDF with respect to the projected points on the image. Suppose $p_i(u_i, v_i)$ denotes the projection of P_i on the image and p'_i represents the coordinates of P_i in the camera coordinate system. Similar to D^{2}IM-Net [29], Laplacian loss L_{Lap} can be formulated as:

$$L_{Lap} = \frac{1}{|P_V|}\sum_{P_i \in P_V} || \triangle g_{suF}(p_i) - l(p_i)||_2^2. \tag{16}$$

The Laplacian of the forward displacement map g_{suF} is:

$$\triangle g_{suF}(p_i) = \frac{\partial^2 g_{suF}(p_i)}{\partial u_i^2} + \frac{\partial^2 g_{suF}(p_i)}{\partial v_i^2}. \tag{17}$$

In case P_i is close to the visible surface of an object, $N(p_i)$ is the unit normal from the ground-truth normal map, equivalent to the gradient of the SDF with respect to p'_i as:

$$N(p_i) = \frac{\partial SDF(P_i)}{\partial p'_i}, P_i \in P_V. \tag{18}$$

Then, the Laplacian of SDF can be defined as:

$$l(p_i) = N(p_i)\frac{\partial p'_i}{\partial^2 u_i} + N(p_i)\frac{\partial p'_i}{\partial^2 v_i}. \tag{19}$$

To enhance the fidelity of the reconstructed object and capture richer small-scale details, ED^{2}IF^2-Net employs a weighted sampling strategy similar to D^{2}IM-Net [29]. This strategy assumes dense sampling of the object, where the density of each sampling point is determined by the number of surrounding sampling points within a specified radius. Inside and outside the object, a clipping policy defines compact sample densities. These densities, along with the same samples, serve as sampling weights during the training of ED^{2}IF^2-Net. The effectiveness of the weighted sampling strategy in reconstructing small-scale details is demonstrated through ablation studies.

4. Experiment Results and Discussion

In Section **??**, the utilized datasets and evaluation metrics are described, while in Section 4.2, the implementation details are outlined. Section 4.3 presents a qualitative and quantitative comparison of ED^{2}IF^2-Net with state-of-the-art methods for single-view implicit 3D reconstruction. Ablation studies are conducted in Section 4.4 to assess the

impact of different factors, and the computational complexity of various methods is analyzed in Section 4.5. Examples showcasing the proposed applications are demonstrated in Section 4.6 and the influence of different camera sensors on ED^{2}IF^2-Net is discussed in Section 4.7.

4.1. Dataset and Metrics

ED^{2}IF^2-Net was trained and tested on a subset of ShapeNet [57], which comprises 13 classes and approximately 44,000 3D models. These models underwent pre-processing using the method proposed by DISN [24] to generate point coordinate–SDF pairs, as well as RGB images and normal maps from 36 random views at a resolution of 224 × 224. For the experiments, we adhered to the official training/validation/testing split.

For the overall quality of the reconstruction, intersection of union (IoU), Chamfer distance (CD) and earth mover distance [58] (EMD) are computed. Moreover, the edge Chamfer distance [59] of the reconstructed shape (ECD-3D) and the edge Chamfer distance in the image [29] (ECD-2D) are used to measure the recovered detail information. The specific definitions of all the above evaluation metrics are as follows:

IoU is used to measure the similarity between the reconstructed object and the ground truth, defined as

$$IoU(PC_P, PC_Q) = \frac{intersection(\Gamma(PC_P), \Gamma(PC_Q))}{union(\Gamma(PC_P), \Gamma(PC_Q))}, \tag{20}$$

where PC_P and PC_Q denote two point clouds, and Γ denotes the operation that converts a point cloud into a voxel grid.

CD is a commonly used metric for measuring the distance between two point clouds, denoted as PC_P, PC_Q, defined as

$$CD(PC_P, PC_Q) = \sum_{p_1 \in PC_P} \min_{p_2 \in PC_Q} ||p_1 - p_2||_2^2 + \sum_{p_2 \in PC_Q} \min_{p_1 \in PC_P} ||p_1 - p_2||_2^2. \tag{21}$$

EMD is a metric frequently used to measure the distance between two point clouds, denoted as PC_P, PC_Q, by considering the distribution problem. It can be defined as

$$EMD(PC_P, PC_Q) = \min_{\phi: PC_P \to PC_Q} \sum_{p \in PC_P} ||p - \phi(p)||_2, \tag{22}$$

where $\phi : PC_P \to PC_Q$ represents a bijection between the two point clouds.

ECD-3D is a metric calculated as the Chamfer distance (CD) between the edge points on the ground-truth object and the reconstructed object. The "edgeness" property of each sampled point from a 3D object is defined as

$$\psi(p_j) = \min_{p_k \in \Omega_j} |n_j \cdot n_k|, \tag{23}$$

where Ω_j represents the set of neighboring points of p_j, and n_j and n_k denote the unit normal vectors at points p_j and p_k, respectively.

In our implementation, we consider a set of 10 neighbouring points (Ω) for each point and we evaluate the edge feature recovery using points with an "edgeness" property ($\psi(p_j)$) value below 0.8.

ECD-2D represents the Chamfer distance (CD) between edge pixels on the rendered images. In our implementation, we utilize the Canny operator to extract the edges from the rendered normal map of the reconstructed object, which has a resolution of 224 × 224, in order to obtain the edge pixels.

4.2. Implementation Details

In ED^{2}IF^2-Net, an RGB image of size 224×224 is used as input, and the model outputs signed distance values of the query points. The iso-surface mesh is visualized using Marching Cubes with a resolution of $128 \times 128 \times 128$. The network is implemented in Pytorch [60] and the training parameters are set as follows: batch_size of 16, Adam optimizer [61] with a learning rate of 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 10^{-5}. During training, 2048 query points are randomly selected based on the weighted sampling strategy for loss calculation and back propagation. The experiments were conducted using PyCharm Community Edition, and the training of ED^{2}IF^2-Net was performed on two Nvidia RTX 3090 graphics cards, taking 1 to 3 days, depending on the specific settings. The hyperparameters ω_0, ω_1, and ω_2 in the deformation loss term L_{Def} of the network loss function were fixed at 0.5, 0.25, and 0.25, respectively. The value of ω_0 at 0.5 was chosen to emphasize the influence of the deformed implicit field g_{st} on the final reconstruction results. The training process involved 500 epochs and the learning rate was adaptively optimized using the Adam optimizer as described above. Further details on the implementation can be found in Appendix A.

4.3. Comparison with SOTA Approaches

Our comparison concentrates on the implicit models that have achieved state-of-the-art results to date, mainly including IM-Net [19], MDISN [25], DISN [24], D^{2}IM-Net [29], and its variant D^{2}IM-Net$_{GL}$. IM-Net is similar to the coarse shape decoder of ED^{2}IF^2-Net, and MDISN and D^{2}IM-Net are the corresponding baselines for the deformation decoder and the surface detail decoder, respectively. Moreover, DISN is by far the most excellent single-view implicit 3D reconstruction method with respect to geometric details. For comparative fairness, the above networks were all trained and tested based on the same benchmarks.

Table 1 presents the quantitative comparison of all the aforementioned methods on ShapeNet. The results indicate that ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L outperform other methods in most object categories, demonstrating significantly higher mean values for each evaluation metric across 13 object categories compared to the other methods. Notably, ED^{2}IF^2-Net-L achieves state-of-the-art quantitative results on ShapeNet, with an IoU of 61.1, CD of 7.26, EMD of 2.51, ECD-3D of 6.08, and ECD-2D of 1.84. When compared to DISN, ED^{2}IF^2-Net-L exhibits a 7% increase in mean IoU, and a 34%, 12%, 12%, and 26% decrease in mean CD, EMD, ECD-3D, and ECD-2D, respectively. These quantitative results demonstrate that both ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L excel not only in overall shape (topological structure) but also in recovering edge details (surface details). It is important to note that ED^{2}IF^2-Net may not achieve the best performance in every category, which could be attributed to the the network being trained on all categories of ShapeNet. The network's sensitivity to the quantity and diversity of models within a single category may result in slightly inferior reconstruction results for categories with fewer models or predominantly similar models, such as phones.

The qualitative comparison of different methods is presented in Figure 6. From the figure, it is evident that IM-Net can only reconstruct the coarse shape of the object, resulting in a loss of significant topological structure details (such as holes of the sofa backrest and handles of the table drawers) as well as surface details (e.g., chair backrest). Compared to IM-Net, DISN performs better in rebuilding topological structures and surface details, although the results may contain geometric noise leading to blurry surfaces (e.g., sofa backrest surface). However, DISN struggles in recovering details at small scales (bottom and backrest of the chair).

While MDISN can reconstruct more detailed topological structures, it fails to recover surface details and even introduces shape distortions to the object (e.g., speaker and table). On the other hand, D^{2}IM-Net shows promise in surface detail recovery but often produces incorrect topological structures for highly curved shapes (e.g., armrests and bottom of the chair) and introduces numerous artifacts (such as the table).

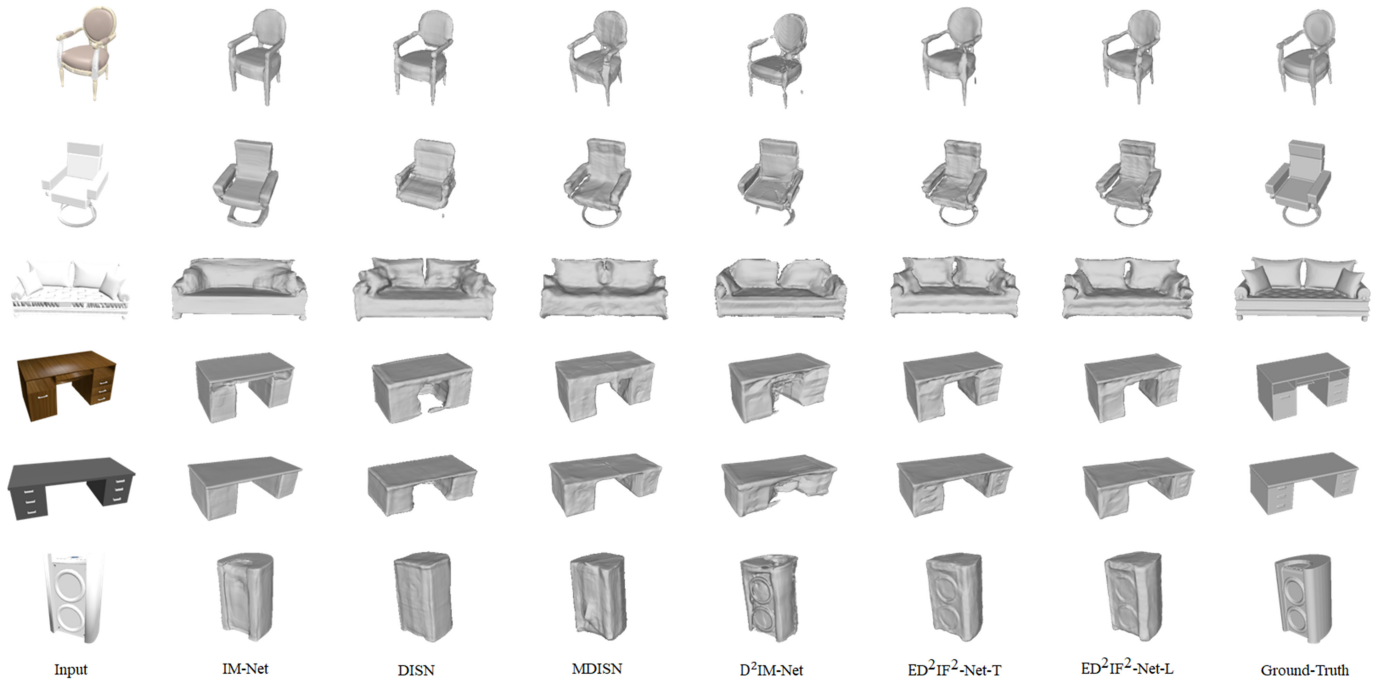

Figure 6. Qualitative comparison of various methods for single-view 3D reconstruction on ShapeNet.

In contrast, both ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L are able to reconstruct more visually appealing qualitative results. These methods enable the reconstruction of more complex topologies (e.g., holes in sofa backrests, handles of table drawers) and capture finer small-scale surface details (e.g., chair backrests). These findings align with the quantitative comparison in Table 1 and validate that ED^{2}IF^2-Net can effectively generate high-fidelity 3D shapes with accurate topological structure and surface details.

Table 1. Quantitative comparison of all methods for single-view 3D reconstructions on ShapeNet. Evaluation metrics include IoU (%, the larger the better), CD ($\times$0.001, the smaller the better), EMD ($\times$100, the smaller the better), ECD-3D ($\times$0.01, the smaller the better), and ECD-2D (the smaller the better). CD and EMD are calculated on 2048 sample points. ECD-3D is computed on 20K points. ECD-2D is calculated on the normal maps with a resolution of 224 $\times$ 224. Top scores are highlighted in **bold** and underlined, while the *italic* one is the second.

		Plane	Bench	Box	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Boat	Mean
IoU↑	IM-Net	55.4	49.5	51.5	74.5	52.2	56.2	29.6	52.6	52.3	64.1	45.0	70.9	56.6	54.6
	DISN	57.5	52.9	52.3	74.3	54.3	56.4	34.7	54.9	59.2	65.9	47.9	72.9	55.9	57.0
	MDISN	60.4	54.6	52.2	74.5	55.6	59.4	38.2	**55.8**	62.2	68.5	48.6	73.5	60.4	58.8
	D^{2}IM-Net	60.6	55.7	52.1	74.6	56.2	61.9	*40.8*	54.5	63.4	69.3	48.2	*73.8*	**62.5**	59.5
	D^{2}IM-Net$_{GL}$	59.2	53.8	52.6	73.5	54.7	62.4	**41.1**	54.3	62.9	68.5	48.0	**74.3**	61.6	59.0
	ED^{2}IF^2-Net-T	*62.9*	*57.8*	*55.2*	*75.8*	*56.5*	*63.7*	38.9	54.6	*64.5*	**71.1**	*49.3*	72.6	61.8	*60.4*
	ED^{2}IF^2-Net-L	**63.5**	**59.6**	**56.5**	**76.4**	**57.3**	**64.2**	39.6	*55.7*	**65.1**	*70.6*	**50.8**	72.1	*62.3*	**61.1**
CD↓	IM-Net	12.65	15.10	11.39	8.86	11.27	13.77	63.84	21.83	8.73	10.30	17.82	7.06	13.25	16.61
	DISN	9.96	8.98	10.19	5.39	7.71	10.23	25.76	17.90	5.58	9.16	13.59	6.40	11.91	10.98
	MDISN	**5.77**	6.29	8.78	5.21	6.68	8.13	15.59	14.54	6.98	6.96	10.36	**5.36**	**6.20**	8.22
	D^{2}IM-Net	7.32	6.03	9.16	4.98	6.41	8.25	14.57	14.69	5.14	6.45	9.83	5.42	7.56	8.14
	D^{2}IM-Net$_{GL}$	7.14	6.15	8.92	5.06	6.34	8.03	14.59	14.41	5.27	6.58	9.67	5.49	7.12	8.06
	ED^{2}IF^2-Net-T	6.31	*5.62*	*8.13*	*4.66*	*6.15*	*7.59*	*14.17*	*13.06*	**4.38**	**6.06**	*8.64*	5.47	6.45	*7.44*
	ED^{2}IF^2-Net-L	*5.89*	**5.34**	**7.86**	**4.52**	**6.03**	**7.42**	**13.91**	**12.75**	*4.41*	*6.12*	**8.54**	*5.39*	*6.23*	**7.26**
EMD↓	IM-Net	2.90	2.80	3.14	2.73	3.01	2.81	5.85	3.80	2.65	2.71	3.39	2.14	2.75	3.13
	DISN	2.67	2.48	3.04	2.67	2.67	2.73	4.38	3.47	2.30	2.62	3.11	2.06	2.77	2.84
	MDISN	2.33	2.17	**2.91**	2.70	**2.52**	*2.50*	3.67	3.30	2.17	2.43	2.81	2.11	2.42	2.62
	D^{2}IM-Net	2.24	2.18	*2.93*	2.61	2.65	2.62	3.72	3.28	*2.14*	**2.36**	2.78	**1.91**	2.53	2.61
	D^{2}IM-Net$_{GL}$	2.32	2.13	3.01	2.58	2.62	2.66	3.67	3.41	2.25	2.44	2.86	2.00	2.49	2.65
	ED^{2}IF^2-Net-T	*2.12*	*2.15*	2.96	*2.57*	2.59	**2.48**	*3.55*	*3.21*	2.18	2.42	*2.72*	2.08	*2.37*	*2.57*
	ED^{2}IF^2-Net-L	**2.07**	**2.12**	*2.93*	**2.45**	*2.54*	2.51	**3.47**	**3.16**	**2.11**	*2.38*	**2.66**	*1.95*	**2.28**	**2.51**

Table 1. *Cont.*

		Plane	Bench	Box	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Boat	Mean
ECD-3D↓	IM-Net	7.89	6.85	8.72	8.72	6.61	8.20	9.95	10.80	6.74	7.90	7.10	7.24	8.23	8.07
	DISN	6.84	5.73	6.97	6.80	5.64	7.65	11.27	10.77	3.50	6.06	6.01	7.08	5.83	6.94
	MDISN	6.32	5.13	6.84	6.87	5.57	7.39	10.06	10.26	3.53	6.29	5.95	6.72	5.94	6.68
	D^2IM-Net	5.67	4.77	6.61	7.28	5.23	6.74	9.18	9.09	3.43	6.42	6.30	6.09	5.68	6.34
	D^2IM-Net$_{GL}$	5.98	5.16	6.91	**6.46**	**5.04**	7.13	**8.97**	9.73	3.57	*6.02*	**5.67**	6.60	**5.34**	6.35
	ED^2IF^2-Net-T	**5.31**	*4.51*	**6.54**	6.76	5.19	*6.51*	*9.07*	*8.94*	*3.16*	6.03	5.93	*6.02*	5.45	*6.11*
	ED^2IF^2-Net-L	*5.33*	**4.45**	*6.60*	*6.72*	*5.15*	**6.49**	9.11	**8.87**	**3.12**	**5.98**	*5.86*	**5.96**	*5.38*	**6.08**
ECD-2D↓	IM-Net	2.53	2.85	4.47	3.34	2.70	3.23	3.36	4.20	3.14	2.98	2.85	2.42	3.05	3.16
	DISN	2.67	2.21	2.25	2.04	1.98	3.16	4.86	3.34	1.35	2.06	2.07	2.26	2.00	2.48
	MDISN	2.36	2.13	2.01	2.12	1.64	2.65	4.47	2.98	1.39	2.08	1.97	1.93	1.91	2.28
	D^2IM-Net	1.99	1.67	1.79	2.07	1.71	1.95	3.16	2.64	*1.28*	2.01	1.88	1.62	1.73	1.96
	D^2IM-Net$_{GL}$	1.98	*1.77*	1.74	**1.77**	**1.58**	2.68	*3.01*	2.72	*1.77*	**1.78**	**1.74**	2.14	2.27	*2.07*
	ED^2IF^2-Net-T	**1.92**	*1.51*	**1.66**	1.94	*1.63*	*1.87*	**2.99**	*2.48*	1.36	2.04	*1.79*	*1.58*	*1.65*	*1.88*
	ED^2IF^2-Net-L	*1.96*	**1.44**	*1.68*	*1.85*	*1.59*	**1.79**	3.04	**2.41**	**1.26**	*1.88*	1.84	**1.52**	**1.63**	**1.84**

4.4. Ablation Studies

To validate the effectiveness of the individual components of ED^2IF^2-Net and the loss functions, extensive qualitative and quantitative ablation studies were carried out. All the networks used in the ablation studies were trained and tested on the chair class of ShapeNet. To be specific, the following network options were designed:

- Option 1: In this option, we keep the original encoder PVT in the network, plus the coarse shape decoder (CSD) and a random sampling strategy, and the loss function L_{Coa} is applied. It can be seen from Figure 7 that the coarse shape decoder and the random sampling strategy can only reconstruct the coarse shape with few structure details and no surface details. It is consistent with the quantitative results in Table 2.
- Option 2: On the basis of the first option, the network is trained with weighted sampling (WS). It can be found from Figure 7 that WS enables the network to reconstruct more details, especially at small scales.
- Option 3: In this option, we still use PVT as the encoder. However, we try to directly initialize a random signed distance value for each query point and iteratively refine it in the deformation decoder (DD). Then, the network is trained only constrained by deformation loss L_{Def} with WS. It can be observed from Figure 7 that the network without the coarse implicit field reconstructs awful surfaces and topologies. Moreover, quite a few surface artifacts emerge due to the absence of the coarse implicit field near the shape.
- Option 4: With this option, CSD together with the DD serve as the decoders and only L_{Coa} with WS is used for the loss estimation. It can be seen in Figure 7 that such a network creates fewer shape artifacts and distortions, but it still fails to reconstruct a full shape of the structure, which is attributed to the fact that the loss function takes no account of the intermediate implicit fields generated in the iterative deformation.
- Option 5: Based on the previous options, L_{Coa} and L_{Def} are applied to train the CSD and the DD, respectively. WS is also used here. From Figure 7, it is illustrated that the network with this option is capable of reconstructing more accurate topological structures and producing a smoother shape.
- Option 6: In this option, the surface detail decoder in ED^2IF^2-Net with WS cancels the prediction of the backward displacement map and the deformed implicit field is only fused with the forward displacement map. The surface detail decoder in this case is represented as SDD_S and the normal case is denoted as SDD_N. It can be noticed from Figure 7 that, without the backward displacement map, the surface details of the results may be incorrectly reconstructed and distortions may occur at the structural level, possibly owing to the lack of the backward displacement map, which prevents fine-tuning.

- Option 7: Only the L_{Lap} of the standard ED^{2}IF^2-Net loss functions is removed and the rest remains unchanged. It can be noted from Figure 7 that, in this case, the surface details of the reconstruction cannot be clearly recovered and may produce distortions.
- Option 8: The encoder in ED^{2}IF^2-Net-T is replaced with ResNet18, keeping the rest of the settings fixed. The reconstruction results are shown in Figure 7 and it can be noticed that there exist plenty of artifacts, which may be caused by ResNet18 being slightly inferior to PVT in terms of feature extraction, proving that PVT is optimal for ED^{2}IF^2-Net.
- Option 9: When this option is selected, all HAMs in the SDD_N of the standard ED^{2}IF^2-Net are removed and the rest of the network settings remain fixed. As shown in Figure 7, the surface details of the reconstructed objects become unclear without the HAM, which is consistent with the quantitative results in Table 2, demonstrating that the variant leads to an increase in ECD-3D and ECD-2D. These results confirm the effectiveness of HAMs in enhancing surface details.
- Option 10: We remove the DD from the standard ED^{2}IF^2-Net and exclude the L_{Def} term from the loss function to create a variant pipeline similar to D^{2}IM-Net. As shown in Figure 7, the shapes reconstructed by this variant are not comparable to the ones reconstructed by the standard ED^{2}IF^2-Net. It is worth noting that the quantitative comparisons in Tables 1 and 2 show that, although the variant (marked in orange) has slightly lower performance than the standard ED^{2}IF^2-Net to some extent, it still outperforms D^{2}IM-Net, which confirms the superiority of our network pipeline.
- Option 11: To further validate the effectiveness of the deformation decoder (DD) and deformation loss L_{Def} in reconstructing finer topological structures, we add the DD to the network of option 10 while keeping the other settings unchanged. As shown in Figure 7, this variant generally reconstructs object shapes with more detailed topological structures compared to option 10. This further demonstrates the contribution of the DD in reconstructing finer topological structures of objects. However, it is worth noting that the variant still struggles to generate visually appealing object shapes compared to the standard ED^{2}IF^2-Net. This observation emphasizes the importance of the deformation loss L_{Def} in the reconstruction process.
- Option 12: To further demonstrate the superiority of the proposed method in feature extraction, we replace the PVT-Tiny and PVT-Large image encoders in the standard ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L with DeiT-Tiny and DeiT-Base [37], respectively, while keeping the other settings unchanged. The qualitative results are presented in Figure 7. It can be observed that when the image encoders of ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L are replaced by DeiT-Tiny and DeiT-Base, respectively, the network tends to reconstruct inferior results, which exhibit poor topological structure and surface details. This further confirms the effectiveness of ED^{2}IF^2-Net in feature extraction.
- Option 13: To further validate the effectiveness of HAM in the surface detail decoder for enhancing surface detail representation, all HAMs in the surface detail decoder of the standard model are replaced with CBAMs [45], while keeping the other settings unchanged. The qualitative reconstruction results are depicted in Figure 7. In comparison to the standard ED^{2}IF^2-Net, the variant encounters challenges in capturing and recovering clear surface details of the object, resulting in the presence of artifacts around the shape. This option further demonstrates that HAM is more effective than CBAM in enhancing the capability of ED^{2}IF^2-Net to handle surface details.
- Option 14: The standard ED^{2}IF^2-Net proposed in this paper, including all components and the loss function with WS.

The visualization and quantitative results of the ablation studies are presented in Figure 7 and Table 2, respectively.

Table 2. Quantitative comparison for ablation studies, where ✓ indicates the component or loss term used by the option. Evaluation metrics remain IoU (%), CD (×0.001), EMD (×100), ECD-3D (×0.01), and ECD-2D. Top scores are highlighted in **bold** and underlined.

	PVT-Tiny	ResNet18	DeiT-Tiny	PVT-Large	DeiT-Base	CSD	DD	SDD_S	SDD_N	HAM	CBAM	WS	L_{Coa}	L_{Def}	L_{Ove}	L_{Lap}	IoU↑	CD↓	EMD↓	ECD-3D↓	ECD-2D↓
Option 1	✓					✓							✓				52.4	9.96	3.11	7.42	3.58
				✓		✓							✓				52.9	9.87	3.09	7.36	3.52
Option 2	✓					✓						✓	✓				53.5	9.23	3.04	7.16	3.17
				✓		✓						✓	✓				53.7	9.31	2.95	6.98	3.06
Option 3	✓						✓					✓		✓			54.1	8.67	2.96	6.85	2.88
				✓			✓					✓		✓			54.2	8.63	2.87	6.64	2.75
Option 4	✓					✓	✓					✓	✓				54.7	8.32	2.89	6.53	2.54
				✓		✓	✓					✓	✓				54.9	8.17	2.81	6.35	2.46
Option 5	✓					✓	✓					✓	✓	✓			55.1	7.74	2.83	6.29	2.27
				✓		✓	✓					✓	✓	✓			55.4	7.63	2.75	6.02	2.11
Option 6	✓					✓	✓	✓		✓		✓	✓	✓	✓	✓	55.6	7.21	2.78	5.87	1.95
				✓		✓	✓	✓		✓		✓	✓	✓	✓	✓	56.1	6.97	2.67	5.64	1.82
Option 7	✓					✓	✓		✓	✓		✓	✓	✓	✓		55.9	6.58	2.71	5.49	1.76
				✓		✓	✓		✓	✓		✓	✓	✓	✓		56.5	6.39	2.62	5.38	1.67
Option 8		✓				✓	✓		✓	✓		✓	✓	✓	✓	✓	51.2	10.83	3.16	7.59	3.65
Option 9	✓					✓	✓		✓			✓	✓	✓	✓	✓	56.4	6.26	2.61	5.21	1.65
				✓		✓	✓		✓			✓	✓	✓	✓	✓	57.0	6.12	2.56	5.18	1.62
Option 10	✓					✓			✓	✓		✓	✓		✓	✓	56.2	6.35	2.64	5.22	1.69
				✓		✓			✓	✓		✓	✓		✓	✓	56.4	6.27	2.61	5.20	1.64
Option 11	✓					✓	✓		✓	✓		✓	✓		✓	✓	56.3	6.30	2.62	5.21	1.67
				✓		✓	✓		✓	✓		✓	✓		✓	✓	56.8	6.14	2.57	5.17	1.61
Option 12			✓			✓	✓		✓	✓		✓	✓	✓	✓	✓	54.8	6.54	2.76	5.43	1.78
					✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	55.6	6.48	2.69	5.32	1.68
Option 13	✓					✓	✓		✓		✓	✓	✓	✓	✓	✓	56.0	6.26	2.72	5.25	1.71
				✓		✓	✓		✓		✓	✓	✓	✓	✓	✓	56.9	6.18	2.64	5.23	1.67
Option 14	✓					✓	✓		✓	✓		✓	✓	✓	✓	✓	56.5	6.15	2.59	5.19	1.63
				✓		✓	✓		✓	✓		✓	✓	✓	✓	✓	**57.3**	**6.03**	**2.54**	**5.15**	**1.59**

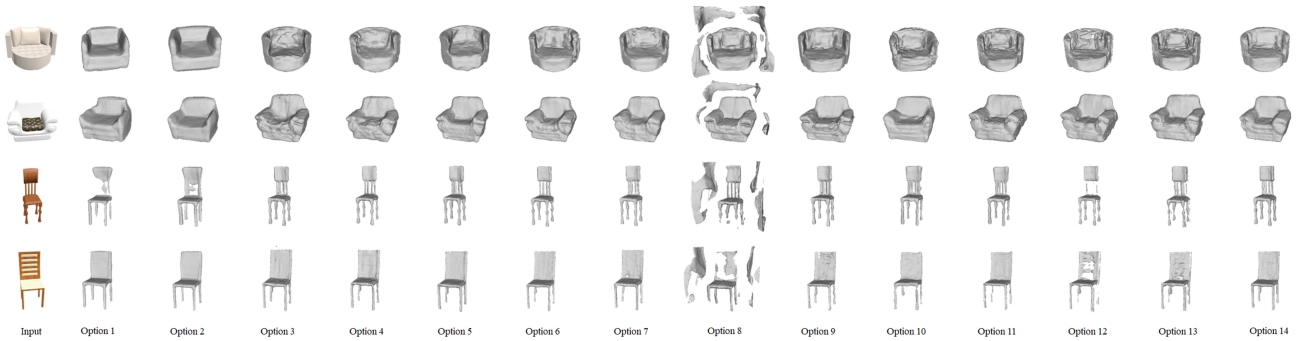

Figure 7. Visualization of the qualitative ablation studies of ED^{2}IF^2-Net-T. It is best viewed magnified on the screen.

Overall, the weighted sampling strategy enables the network to reconstruct small-scale details effectively. Additionally, the deformation decoder, which refines the coarse implicit field, plays a crucial role in capturing the object's topology. The deformation decoder performs optimally when trained with the deformation loss term L_{Def}, while the coarse shape decoder benefits from the L_{Coa}. The deformed implicit field, derived from the deformation decoder, serves as a solid foundation for reconstructing the object's surface, which is further fused with the forward displacement map generated by the surface detail decoder trained by L_{Lap} to recover the surface details of the object. Moreover, the backward displacement map from the surface detail decoder compensates for the deformed implicit field, ensuring the correct topology reconstruction. Furthermore, compared to using ResNet18 as an encoder, the standard ED^{2}IF^2-Net achieves higher-fidelity results. Importantly, the presence of the deformation decoder and the utilization of the deformation loss term L_{Def} contribute to the reconstruction of ED^{2}IF^2-Net with finer topological structures. Furthermore, the PVT architecture, which generates multi-scale hierarchical local features, is more suitable as an image encoder for ED^{2}IF^2-Net compared to other conventional transformers such as DeiT. Lastly, in the surface detail decoder, the HAM module proves to be more effective in improving the model's performance in recovering surface details and ensuring the reconstruction of a correct topological structure compared to CBAM.

The proposed ED^{2}IF^2-Net is inherently superior to D^{2}IM-Net. Specifically, ED^{2}IF^2-Net significantly improves the network's ability to extract features by using PVT instead of ResNet18. Moreover, ED^{2}IF^2-Net iteratively refines the coarse implicit field via the deformation decoder with L_{Def} to reconstruct finer topological structure details of the object, and employs HAM to enhance surface details instead of predicting only the coarse implicit field and the ordinary displacement field as in D^{2}IM-Net. Finally, when the deformation decoder with L_{Def} in the standard ED^{2}IF^2-Net are abolished, the quantitative results achieved by the network still outperform D^{2}IM-Net.

4.5. Computational Complexity

In addition to the qualitative and quantitative experiments described above, we also provide the computational complexity of the various methods in Table 3, specifically in terms of training time and inference time. To ensure a fair comparison, all models were trained and tested using the same settings.

As shown in the table, ED^{2}IF^2-Net-T achieves the fastest training speed, with a training time of 47 h. Similarly, ED^{2}IF^2-Net-L has a relatively shorter training time of 66 h compared to most other models. In terms of inference time, both ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L outperform other methods, with inference times of 97.64 ms and 144.09 ms, respectively.

The above comparison of computational complexity highlights the advantages of the proposed ED^{2}IF^2-Net in terms of faster training speed and shorter inference time.

Table 3. Training time and inference time. All training times as well as inference times are obtained with the same settings, where the inference times are tested with a batch_size of 1.

Model	IM-Net	DISN	MDISN	D^{2}IM-Net	ED^{2}IF^2-Net-T	ED^{2}IF^2-Net-L
Training Time (h)	138	105	84	68	47	66
Inference Time (ms)	204.15	188.19	162.73	146.57	97.64	144.09

4.6. Applications

4.6.1. Test on Online Product Images

ED^{2}IF^2-Net, after being trained on the rendered RGB images, allows for further testing of online product images without ground-truth shapes. The qualitative reconstruction results of ED^{2}IF^2-Net for online product images are presented in Figure 8. This application demonstrates the generalization capability of ED^{2}IF^2-Net.

Figure 8. Examples of reconstruction from online images through ED^{2}IF^2-Net.

4.6.2. Surface Detail Transfer

Surface detail transfer is defined as the fusion of the disentangled enhanced displacement field of a source object with the deformed implicit field of another target object. In this application, the specified surface details can be transferred and Figure 9 shows examples of surface detail transfer between different objects.

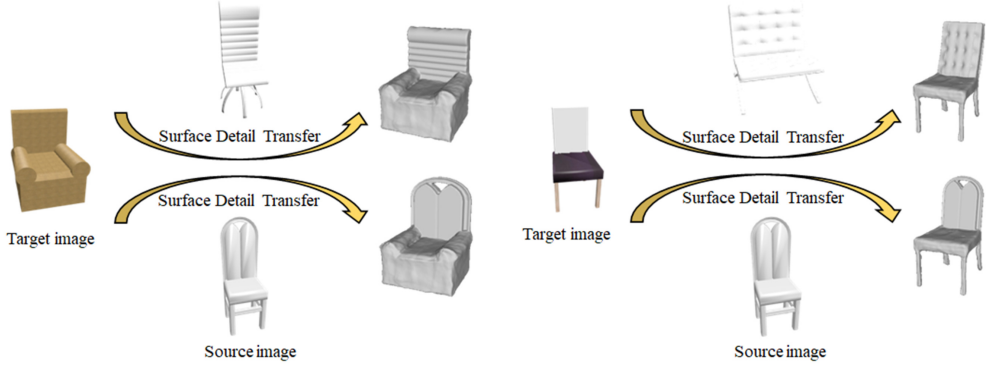

Figure 9. Two examples of surface detail transfer using ED^{2}IF^2-Net, where the backrest details of the source chair are transferred.

4.6.3. Pasting a Logo

We propose that a logo can be pasted on the target object image and then the modified image is used to generate a model with the logo. Figure 10 shows examples of pasting a logo on a model.

Figure 10. Examples about pasting a logo using ED^{2}IF^2-Net.

Actually, D^{2}IM-Net [29] provides similar applications. A quantitative comparison of the different applications of D^{2}IM-Net and ED^{2}IF^2-Net is shown in Table 4. It should be noted that, as there are no ground-truth models for the generated objects, the corresponding ground-truth models are created for the shown generated objects by traditional manual modeling, and the mean values of various evaluation metrics obtained by both D^{2}IM-Net and ED^{2}IF^2-Net are computed. It can be observed from Table 4 that the proposed ED^{2}IF^2-Net achieves more promising performance compared to D^{2}IM-Net in downstream applications, where ED^{2}IF^2-Net-L reaches state-of-the-art performance in these applications, further illustrating the superiority of ED^{2}IF^2-Net over D^{2}IM-Net.

Table 4. Quantitative results of D^{2}IM-Net and ED^{2}IF^2-Net for various applications. Evaluation metrics also include IoU (%), CD ($\times$0.001), EMD ($\times$100), ECD-3D ($\times$0.01), and ECD-2D. The best results for each application are highlighted in **bold** and underlined, while the *italic* one is the second.

		IoU↑	CD↓	EMD↓	ECD-3D↓	ECD-2D↓
Surface Detail Transfer	D^{2}IM-Net	51.4	6.87	3.01	5.86	2.25
	ED^{2}IF^2-Net-T	*52.3*	*6.69*	*2.88*	*5.72*	*2.01*
	ED^{2}IF^2-Net-L	**53.1**	**6.58**	**2.75**	**5.65**	**1.92**
Pasting a Logo	D^{2}IM-Net	53.4	6.67	2.93	5.66	2.08
	ED^{2}IF^2-Net-T	*54.6*	*6.52*	*2.84*	*5.57*	*1.88*
	ED^{2}IF^2-Net-L	**55.8**	**6.36**	**2.69**	**5.42**	**1.75**

4.7. Discussion about the Effects of Camera Sensor Type on ED^{2}IF^2*-Net*

In the previous experiments and applications, the images utilized were acquired using a standard camera sensor model, which allowed for capturing images without significant distortion. However, in various industries such as drone aerial photography, security surveillance, and automotive, wide-angle and fisheye imaging sensors are extensively employed. These sensors typically have a field of view (FOV) greater than 100 degrees, which is considerably larger compared to standard camera sensors. Hence, in this section, we primarily focus on discussing the effects of images captured by wide-angle and fisheye imaging sensors on the performance of ED^{2}IF^2-Net.

There are existing works [62–64] that utilize images captured by wide-angle or fisheye sensors for 3D reconstruction and other related tasks. For instance, Ma et al. [62] proposed a specific model for fisheye sensors and introduced sparse and dense multi-view 3D reconstruction methods based on this model. Strecha et al. [63] performed 3D reconstruction using images captured by fisheye sensors and standard lens models, respectively, employing the Pix4Dmapper software. Kakani et al. [64] proposed a self-calibration method for wide-angle and fisheye cameras to correct the captured images, allowing for their utilization in 3D reconstruction and other tasks.

In general, wide-angle sensors can capture images with a larger field of view compared to standard lenses, but they often introduce perspective distortion. This distortion can alter the shape of objects in the image, making it challenging for ED^{2}IF^2-Net trained on images acquired from the standard lens model to reconstruct high-fidelity object shapes. On the other hand, fisheye camera sensors can capture images with an extremely wide field of view but introduce barrel distortion, which causes even more severe distortion of objects in the image. Consequently, ED^{2}IF^2-Net faces difficulties in reconstructing accurate object shapes from such distorted images.

To mitigate the effects of perspective distortion and barrel distortion caused by wide-angle and fisheye camera sensors on ED^{2}IF^2-Net, pre-processing techniques such as camera calibration [64] and image correction [65] can be employed to reduce the degree of image distortion. Another approach is to consider training ED^{2}IF^2-Net on publicly available datasets of images captured by wide-angle sensor models and fisheye sensor models, enabling the network to learn about the different distortions using its powerful feature extraction and learning capabilities.

5. Limitations and Future Works

The proposed method has two main limitations. Firstly, although the surface detail decoder enhances surface information, some reconstructed object shapes, such as the speaker in Figure 6, lack prominent surface detail. This limitation may be attributed to the introduction of redundant local features during the implicit field deformation procedure. To address this, future studies should explore adaptive neglect of unnecessary local features as an attractive direction for improvement. Secondly, while ED^{2}IF^2-Net outperforms similar methods in terms of inference speed and performance, it is not specifically designed for real-time 3D reconstruction. This may pose challenges for systems that require real-time reconstruction. To tackle this issue, we plan to leverage a sparse sphere rendering algorithm [33,66] to accelerate inference speed. Additionally, we aim to explore more advanced transformers, such as Swin Transformer V2 [67], to enhance the feature extraction capability of ED^{2}IF^2-Net.

In future work, we will optimize the proposed framework for embedded platforms, considering the following aspects: (1) reducing model parameters and computational complexity by minimizing the number of layers in the Pyramid Vision Transformer or reducing the number of channels in the deformation decoder's convolutional layer while maintaining performance; (2) improving the readout speed of implicit fields and displacement fields by utilizing more efficient data structures, such as hash tables, for data storage; (3) optimizing the training and prediction process through techniques such as distillation [68]; (4) deploying the framework on native embedded platforms to reduce communication and latency.

6. Conclusions

In this paper, we introduce ED^{2}IF^2-Net, the first single-view 3D reconstruction network based on the Pyramid Vision Transformer. Our network disentangles objects' implicit fields into deformed implicit fields and enhanced displacement fields. IFDBs refine the coarse implicit fields by analyzing pixel-aligned local features across scales, capturing finer topological structure details in the deformed implicit fields. Moreover, we enhance the displacement fields in both spatial and channel dimensions to preserve surface details.

By employing a novel deformation loss and Laplacian loss, ED^{2}IF^2-Net achieves high-fidelity reconstruction, capturing both the structure and surface details of objects. On the ShapeNet dataset, ED^{2}IF^2-Net delivers superior performance, with ED^{2}IF^2-Net-L achieving the best mean IoU, CD, EMD, ECD-3D, and ECD-2D values of 61.1, 7.26, 2.51, 6.08, and 1.84, respectively.

Compared to other methods, ED^{2}IF^2-Net excels in reconstructing finer topological structures while preserving enhanced surface details. It overcomes the limitations of alternative approaches that may compromise surface details or yield incorrect topology, resulting in higher-quality reconstructions.

Our research represents a significant milestone in single-view implicit 3D reconstruction. We propose the first transformer-based single-view implicit 3D reconstruction network, opening up new possibilities for solving such tasks using transformers. ED^{2}IF^2-Net achieves state-of-the-art performance on the ShapeNet dataset while maintaining competitive inference time. The proposed IFDB and deformation loss can be readily applied to future works, enabling better reconstruction results in single-view implicit 3D reconstruction. The disentangled deformed implicit fields and enhanced displacement fields in our

network benefit downstream applications, including surface detail transfer and pasting a logo. Furthermore, our framework can be optimized for embedded platforms, shedding new light on industrial applications such as VR/AR. Beyond real-time rendering challenges, the framework holds promise for industries such as robotics and autonomous driving.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z. and X.Y. (Xinsheng Yao); software, X.Y. (Xinsheng Yao); validation, X.Z. and X.Y. (Xinsheng Yao); formal analysis, J.Z. (Junjie Zhang), M.Z., L.Y., X.Y. (Xiaosong Yang), J.Z. (Jianjun Zhang) and H.Z.; investigation, X.Y. (Xinsheng Yao); resources, X.Z.; data curation, X.Y. (Xinsheng Yao); writing—original draft preparation, X.Y. (Xinsheng Yao); writing—review and editing, X.Z., X.Y. (Xinsheng Yao), J.Z. (Junjie Zhang), M.Z., L.Y., X.Y. (Xiaosong Yang), J.Z. (Jianjun Zhang) and H.Z.; visualization, X.Y. (Xinsheng Yao); supervision, X.Z.; project administration, D.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ShapeNet dataset and the Online Products dataset used in this study are available at https://shapenet.org (accessed on 25 May 2023) and ftp://cs.stanford.edu/cs/cvgl/Stanford_Online_Products.zip (accessed on 25 May 2023).

Acknowledgments: Our code is built on top of D^{2}IM-Net and MDISN, and we are very grateful to their authors for their crucial support of the open source release of the code.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

All the abbreviations in the manuscript and their respective explanations are as follows:

CNN	Convolutional Neural Network
MLP	Multi-Layer Perceptron
PVT	Pyramid Vision Transformer
IFDB	Implicit Field Deformation Block
HAM	Hybrid Attention Module
SDF	Signed Distance Function
SRA	Spatial-Reduction Attention
SOTA	State Of The Art
IoU	Intersection of Union
CD	Chamfer Distance
EMD	Earth Mover Distance
ECD-3D	Edge Chamfer Distance of the reconstructed shape
ECD-2D	Edge Chamfer Distance in the image
FOV	Field of View
CSD	Coarse Shape Decoder
WS	Weighted Sampling
DD	Deformation Decoder
SDD_N	Normal Surface Detail Decoder
SDD_S	Surface Detail Decoder predicting only a single forward displacement map

Appendix A. Implementation Details

The implementation details mentioned in Section 4.2 of the main text are explained here.

Appendix A.1. Advantages of Utilizing a 224 × 224 *RGB Image as Input*

The reasons for choosing an RGB image with a resolution of 224 × 224 as the network input are as follows:

- Dataset compatibility: The majority of images in existing publicly available 3D reconstruction datasets are based on a resolution of 224 × 224. Therefore, selecting RGB images with a resolution of 224 × 224 as input ensures better alignment with the dataset, leading to improved training efficacy of the network.
- Resource constraints: Higher resolution images as input increase computational and memory requirements, resulting in longer training times and higher hardware demands. By opting for RGB images with a resolution of 224 × 224 as input, computational resource consumption is reduced while maintaining higher performance levels.
- Information preservation: 3D reconstruction involves processing and analyzing input images to extract relevant features. By choosing RGB images with a resolution of 224 × 224 as input, more detailed information can be preserved, resulting in enhanced 3D reconstruction performance.

Appendix A.2. Analysis of Parameter Settings

Appendix A.2.1. Setting batch_size as 16

Setting a smaller value for 'batch_size' can yield the following advantages:

- Reduced memory consumption: A smaller batch_size leads to decreased memory usage since fewer data samples need to be stored per batch. This enables a larger number of batches to fit within the available memory, facilitating efficient training and inference processes.
- Improved model stability: A smaller batch_size enhances the stability of the model by introducing greater randomness in the samples within each batch. This randomization can help mitigate the risk of overfitting, resulting in a more robust and generalizable model.
- Improved tuning effectiveness: A smaller batch_size allows for faster observation of the model's training progress. This expedited feedback loop enables quicker adjustments and fine-tuning of hyperparameters.

Furthermore, the model's performance was compared for various batch_sizes and the corresponding quantitative results are displayed in Figure A1. It should be noted that ED^{2}IF^2-Net-L was not trained with a batch_size of 32 due to memory limitations. From the figure, it can be observed that both ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L achieve the best performance when the batch_size is set to 16. However, it is worth noting that these models also exhibit the highest memory utilization among the tested batch_sizes.

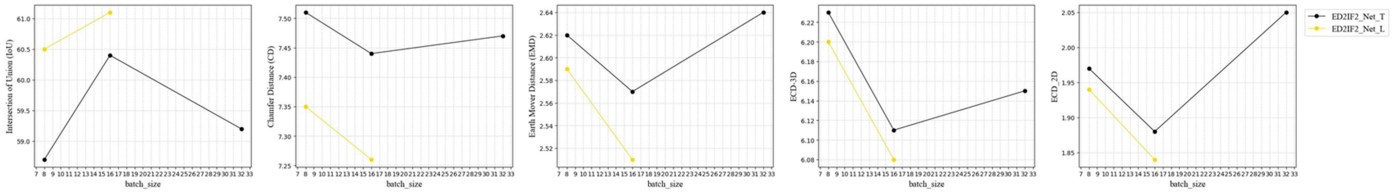

Figure A1. Performance comparison of models with different batch_size; other settings remain fixed. Evaluation metrics include IoU, CD, EMD, ECD-3D, and ECD-2D.

Appendix A.2.2. Setting Learning Rate as 5×10^{-5}

The learning rate should be carefully selected in conjunction with the batch_size to achieve optimal performance. ED^{2}IF^2-Net is trained using different learning rates for a batch_size of 16. We consider three main learning rates: 1×10^{-4}, 5×10^{-5}, and 1×10^{-5}, and compare the performance of the models trained with these different learning rates. Figure A2 illustrates the performance of ED^{2}IF^2-Net-T and ED^{2}IF^2-Net-L under different learning rates. It can be observed that the optimal model performance is achieved with a learning rate of 5×10^{-5} and a batch_size of 16.

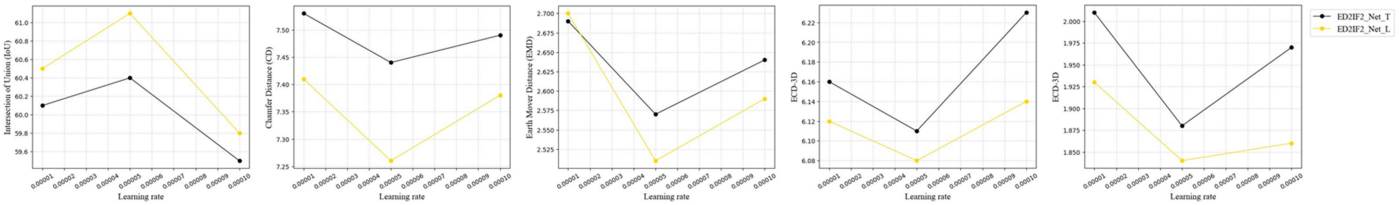

Figure A2. Performance of the models with different learning rates is compared with batch_size set to 16 and other settings kept fixed. Evaluation metrics are IoU, CD, EMD, ECD-3D, and ECD-2D.

Appendix A.2.3. Reasons for Other Settings

The Adam optimizer is a widely used algorithm for adaptive learning rate optimization. It combines the benefits of RMSProp and Adagrad [69] with bias correction, effectively addressing issues such as gradient disappearance and explosion. Adam offers advantages such as self-adaptive learning rate, fast convergence, robustness, and efficient memory consumption. In our experiments, we adopt the default settings of the Adam optimizer, specifically $\beta_1 = 0.9$ and $\beta_2 = 0.999$. This configuration has demonstrated excellent performance across a wide range of experiments.

Weight decay is a technique that mitigates model complexity by introducing a penalty term to the loss function, thereby enhancing the model's generalization capability. A weight decay value of 10^{-5} has shown consistent effectiveness across numerous models. Additionally, this value strikes a balance as it effectively combats overfitting without significantly compromising model performance. Hence, a weight decay of 10^{-5} is considered a suitable and reasonable choice in our experiments.

The choice of sampling 2048 query points strikes a balance between computational efficiency and model performance. When the number of query points is small, the network may struggle to acquire sufficient knowledge, resulting in poor reconstruction performance. Conversely, an excessively large number of query points significantly increases computational costs and slows down network training. Previous studies [24–26,29] have validated that 2048 query points offer an appropriate compromise. This number ensures that the network captures ample information while maintaining manageable computational overhead.

References

1. Zai, S.; Zhao, M.; Yiran, X.; Yunpu, M.; Roger, W. 3D-RETR: End-to-End Single and Multi-View3D Reconstruction with Transformers. In Proceedings of the British Machine Vision Conference (BMVC), Virtual, 22–25 November 2021 ; British Machine Vision Association: Durham, UK, 2021; p. 405.
2. Peng, K.; Islam, R.; Quarles, J.; Desai, K. Tmvnet: Using transformers for multi-view voxel-based 3d reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 18–24 June 2022; pp. 222–230.
3. Yagubbayli, F.; Tonioni, A.; Tombari, F. LegoFormer: Transformers for Block-by-Block Multi-view 3D Reconstruction. *arXiv* **2021**, arXiv:2106.12102.
4. Tiong, L.C.O.; Sigmund, D.; Teoh, A.B.J. 3D-C2FT: Coarse-to-fine Transformer for Multi-view 3D Reconstruction. In Proceedings of the Asian Conference on Computer Vision (ACCV), AFCV, Macau, China, 4–8 December 2022; pp. 1438–1454.
5. Li, X.; Kuang, P. 3D-VRVT: 3D Voxel Reconstruction from A Single Image with Vision Transformer. In Proceedings of the 2021 International Conference on Culture-Oriented Science & Technology (ICCST), IEEE, Beijing, China, 18–21 November 2021; pp. 343–348.
6. Xie, H.; Yao, H.; Sun, X.; Zhou, S.; Zhang, S. Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2690–2698.
7. Choy, C.B.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 628–644.
8. Sun, Y.; Liu, Z.; Wang, Y.; Sarma, S.E. Im2Avatar: Colorful 3D Reconstruction from a Single Image. *arXiv* **2018**, arXiv:1804.06375.
9. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 22–29 October 2017; pp. 2107–2115.

10. Wu, J.; Wang, Y.; Xue, T.; Sun, X.; Freeman, W.T.; Tenenbaum, J.B. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
11. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
12. Lun, Z.; Gadelha, M.; Kalogerakis, E.; Maji, S.; Wang, R. 3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks. In Proceedings of the 2017 International Conference on 3D Vision (3DV), IEEE, Qingdao, China, 10–12 October 2017; pp. 67–77.
13. Kurenkov, A.; Ji, J.; Garg, A.; Mehta, V.; Gwak, J.; Choy, C.; Savarese, S. DeformNet: Free-Form Deformation Network for 3D Shape Reconstruction from a Single Image. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE,Lake Tahoe, NV, USA, 12–15 March 2018; pp. 858–866.
14. Lin, C.H.; Kong, C.; Lucey, S. Learning efficient point cloud generation for dense 3d object reconstruction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2018.
15. Kar, A.; Tulsiani, S.; Carreira, J.; Malik, J. Category-specific object reconstruction from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 7–12 June 2015; pp. 1966–1974.
16. Li, X.; Liu, S.; Kim, K.; De Mello, S.; Jampani, V.; Yang, M.H.; Kautz, J. Self-supervised single-view 3d reconstruction via semantic consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 677–693.
17. Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.G. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 55–71.
18. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 15–20 June 2019; pp. 165–174.
19. Chen, Z.; Zhang, H. Learning Implicit Fields for Generative Shape Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 15–20 June 2019; pp. 5932–5941.
20. Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; Geiger, A. Occupancy Networks: Learning 3D Reconstruction in Function Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 15–20 June 2019; pp. 4460–4470.
21. Littwin, G.; Wolf, L. Deep Meta Functionals for Shape Representation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1824–1833.
22. Michalkiewicz, M.; Pontes, J.K.; Jack, D.; Baktashmotlagh, M.; Eriksson, A. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv* **2019**, arXiv:1901.06802.
23. Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; Chen, B. PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 13–19 June 2020; pp. 829–838.
24. Xu, Q.; Wang, W.; Ceylan, D.; Mech, R.; Neumann, U. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 492–502.
25. Wang, Y.; Zhuang, Y.; Liu, Y.; Chen, B. MDISN: Learning multiscale deformed implicit fields from single images. *Vis. Inform.* **2022**, *6*, 41–49. [CrossRef]
26. Xu, Y.; Fan, T.; Yuan, Y.; Singh, G. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 248–263.
27. Bian, W.; Wang, Z.; Li, K.; Prisacariu, V.A. Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image. In Proceedings of the British Machine Vision Conference (BMVC), British Machine Vision Association, Virtual, 22–25 November 2021.
28. Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; Geiger, A. Convolutional Occupancy Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 523–540.
29. Li, M.; Zhang, H. d^2im-net: Learning detail disentangled implicit fields from single images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 20–25 June 2021; pp. 10246–10255.
30. Saito, S.; Huang, Z.; Natsume, R.; Morishima, S.; Kanazawa, A.; Li, H. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2304–2314.
31. Saito, S.; Simon, T.; Saragih, J.; Joo, H. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 13–19 June 2020; pp. 84–93.
32. He, T.; Collomosse, J.; Jin, H.; Soatto, S. Geo-PIFu: Geometry and Pixel Aligned Implicit Functions for Single-view Human Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 9276–9287.

33. Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; Fidler, S. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 20–25 June 2021; pp. 11358–11367.
34. Deng, Y.; Yang, J.; Tong, X. Deformed Implicit Field: Modeling 3D Shapes with Learned Dense Correspondence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 20–25 June 2021; pp. 10286–10296.
35. Yang, M.; Wen, Y.; Chen, W.; Chen, Y.; Jia, K. Deep optimized priors for 3d shape modeling and reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 20–25 June 2021; pp. 3269–3278.
36. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021.
37. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.
38. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 213–229.
39. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844.
40. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, BC, Canada, 11–17 October 2021; pp. 7242–7252.
41. Li, Y.; Wu, C.Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. MViTv2: Improved multiscale vision transformers for classification and detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 18–24 June 2022; pp. 4804–4814.
42. Wang, D.; Cui, X.; Chen, X.; Zou, Z.; Shi, T.; Salcudean, S.; Wang, Z.J.; Ward, R. Multi-view 3d reconstruction with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, BC, Canada, 11–17 October 2021; pp. 5722–5731.
43. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578.
44. Li, G.; Fang, Q.; Zha, L.; Gao, X.; Zheng, N. HAM: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recognit.* **2022**, *129*, 108785. [CrossRef]
45. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
46. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 21–26 July 2017; pp. 624–632.
47. Tang, Y.; Gong, W.; Chen, X.; Li, W. Deep inception-residual Laplacian pyramid networks for accurate single-image super-resolution. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1514–1528. [CrossRef]
48. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2015.
49. Li, S.; Xu, X.; Nie, L.; Chua, T.S. Laplacian-Steered Neural Style Transfer. In Proceedings of the 25th ACM international conference on Multimedia. Association for Computing Machinery, Mountain View, CA, USA, 23–27 October 2017; pp. 1716–1724.
50. Liu, S.; Li, T.; Chen, W.; Li, H. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7708–7717.
51. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)* Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
52. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
54. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Comput. Graph.* **1987**, *21*, 163–169. [CrossRef]
55. Esedog, S.; Ruuth, S.; Tsai, R. Diffusion generated motion using signed distance functions. *J. Comput. Phys.* **2010**, *229*, 1017–1042. [CrossRef]

56. Yao, Y.; Schertler, N.; Rosales, E.; Rhodin, H.; Sigal, L.; Sheffer, A. Front2back: Single view 3d shape reconstruction via front to back prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 13–19 June 2020; pp. 531–540.
57. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012.
58. Remelli, E.; Lukoianov, A.; Richter, S.; Guillard, B.; Bagautdinov, T.; Baque, P.; Fua, P. MeshSDF: Differentiable Iso-Surface Extraction. In *Advances in Neural Information Processing Systems (NeurIPS)*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 22468–22478.
59. Chen, Z.; Tagliasacchi, A.; Zhang, H. Bsp-net: Generating compact meshes via binary space partitioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 13–19 June 2020; pp. 45–54.
60. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
61. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
62. Ma, C.; Shi, L.; Huang, H.; Yan, M. 3d reconstruction from full-view fisheye camera. *arXiv* **2015**, arXiv:1506.06273.
63. Strecha, C.; Zoller, R.; Rutishauser, S.; Brot, B.; Schneider-Zapp, K.; Chovancova, V.; Krull, M.; Glassey, L. Quality assessment of 3D reconstruction using fisheye and perspective sensors. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 215. [CrossRef]
64. Kakani, V.; Kim, H.; Kumbham, M.; Park, D.; Jin, C.B.; Nguyen, V.H. Feasible Self-Calibration of Larger Field-of-View (FOV) Camera Sensors for the Advanced Driver-Assistance System (ADAS). *Sensors* **2019**, *19*, 3369. [CrossRef]
65. Fan, J.; Zhang, J.; Maybank, S.J.; Tao, D. Wide-angle image rectification: A survey. *Int. J. Comput. Vis.* **2022**, *130*, 747–776. [CrossRef]
66. Hart, J.C. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *Vis. Comput.* **1996**, *12*, 527–545. [CrossRef]
67. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin Transformer V2: Scaling Up Capacity and Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 18–24June 2022; pp. 12009–12019.
68. Alkhulaifi, A.; Alsahli, F.; Ahmad, I. Knowledge distillation in deep learning and its applications. *PeerJ Comput. Sci.* **2021**, *7*, e474. [CrossRef]
69. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.