Gottfried Wilhelm Leibniz Universität Hannover
Institute of Photogrammetry and GeoInformation

Implicit Semantic Scene Surface Reconstruction from Point Clouds

In the course Geodesy and Geoinformation

Master Thesis

of

Wenhao Cai

Examiner:
Prof. Dr.-Ing. habil. Christian Heipke
Superviser:
Dr.-Ing. Max Mehltretter

Hannover, August 2024

Statement

I declare that this thesis has been composed solely by myself under the guidance of my supervisor. It has not been submitted, in whole or in part, to any examination authority. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

____

Signature

____

Place, Date

Abstract

In the realm of semantic scene surface reconstruction research, which involves the dual tasks of 3D reconstruction and semantic segmentation, effectively leveraging the synergistic information between these two tasks has remained a persistent challenge. Traditional deep learning based approaches that sequentially address these tasks often result in inconsistencies between geometric and semantic boundaries. Recently, implicit functions have gained widespread application in this domain. By integrating the strengths of occupancy functions, semantic occupancy networks have been introduced to concurrently perform 3D reconstruction and semantic segmentation. To address the performance bottleneck of this network, this paper proposes structural optimizations, mainly in feature encoding module, for the semantic occupancy network. Additionally, it introduces new reweighting factors in the hierarchical loss function to tackle the issue of class imbalance inherent in semantic segmentation tasks. The effectiveness of the proposed structural optimizations and the improved performance on minority classes are quantitatively evaluated through a series of experiments.

Keywords Semantic Scene Reconstruction; Deep Implicit Function; Class Imbalance;

Contents

1 Introduction

Three-dimensional(3D) reconstruction and semantic scene understanding are fundamental tasks within computer vision field, comprising the dual objectives of reconstructing geometric surfaces and performing semantic segmentation. This process is crucial across various domains, including computer-aided design, computer animation, virtual reality, medical imaging, autonomous driving and etc.. One of the primary challenge lies in accurately reconstructing surfaces from diverse data modalities, such as noisy point clouds, images, and depth maps, each presenting unique difficulties. Noisy point clouds, for instance, provide scattered spatial data points that often lack the necessary density and continuity to form coherent surfaces, complicating both geometric reconstruction and semantic labeling.

Semantic scene reconstruction involves creating a 3D representation of a scene while assigning semantic labels to different elements within the scene. Traditional methods typically involve explicit geometric modeling combined with segmentation algorithms. Classical pipelines might utilize point cloud denoising followed by surface fitting, while modern approaches increasingly employ deep learning techniques. Methods based on Convolutional Neural Networks (CNNs)(57) and Graph Neural Networks (GNNs)(78) have been adapted for this purpose, demonstrating significant advancements in handling the intricacies of semantic scene reconstruction.

The introduction of implicit functions, particularly signed distance functions (SDF) and occupancy functions, has significantly enhanced the capabilities of semantic scene reconstruction. These implicit representations offer continuous, differentiable models that can accurately capture complex geometries and semantic labels from noisy point clouds. Notable approaches include NeRF(60) (Neural Radiance Fields), which utilizes differentiable volume rendering, methods like Deep Implicit Moving Least-Squares Functions(49), which directly learn from noisy inputs to produce high-fidelity reconstructions, and Points2Surf(24) framework, which learns implicit surfaces directly from raw scans without needing normals. These innovations address key issues such as noise robustness and the reconstruction of fine details and intricate structures, some of them even highly suitable for noisy data scenarios(56).

Despite these advancements, current methods still face several limitations. Challenges include handling high noise levels in point clouds, achieving consistent semantic labeling across varying input densities, ensuring consistency of geometric and semantic boundaries and managing computational complexity. Methods such as SNI-SLAM(109), which integrates semantic information into neural implicit SLAM systems, show promise but also highlight the ongoing difficulties in balancing accuracy, efficiency, and robustness. S3CNet(14), by introducing a neural network architecture based on sparse convolutions, successfully addresses the challenges posed by sparse point clouds in large-scale outdoor scenes, unlike previous attempts focused on small dense indoor environments. However, S3CNet still relies on a sequential execution of binary sub-tasks. To overcome this, the Semantic Occupancy Network(58) introduces an innovation by simultaneously achieving surface reconstruction and semantic segmentation through the integration of occupancy information. Moreover, since previous public datasets like SemanticKITTI(3) and nuScenes(7) mostly provide semantic labels at the voxel level, which are not suitable for learning methods based on implicit surface representations, we utilize a variant of the Hessigheim(39) dataset proposed in the paper of the Semantic Occupancy Network.

In this paper, we attempt to make the following contributions:

- Combining the PointNet based variant and subsequent researched modules built upon it into the Semantic Occupancy Network(58) to optimize the feature encoding structure to achieve improved accuracy in both aspects of geometric and semantic part.

- Introducing novel reweighting strategies into hierarchical loss function to optimize the class imbalance problem in simultaneous geometric and semantic tasks to improve the segmentation accuracy of minority categories.

- Extensive experiments to validate the proposed framework, demonstrating improvements in both geometric and semantic accuracy.

In the subsequent sections of this thesis, the theoretical foundations are firstly established in Chapter 2, including certain aspects like the basic knowledge of the implicit function, semantic scene reconstruction, point feature encoding and the class imbalance problem. Chapter 3 provides a comprehensive survey of the current status of the researches pertinent to the domain of interest. A detailed description of the proposed general framework as well as the corresponding components are given in Chapter 4. The experimental settings are covered in Chapter 5, followed by the detailed illustrations and discussions of the obtained results in Chapter 6. The conclusions and a forward-looking perspective on potential avenues for future investigations is provided in Chapter 7.

2 Theoretical Background

2.1 Implicit Function

Implicit functions are mathematical constructs defined by equations where the dependent and independent variables are implicitly interrelated rather than explicitly expressing the dependent variable as a function of the independent variable. A classical example is the equation of a circle, $x^2 + y^2 = r^2$, which implicitly defines the relationship between x and y. Unlike explicit functions such as $y = f(x)$, implicit functions are more versatile and powerful in representing complex geometries and multivariable relationships.

Implicit functions find extensive applications across various domains: In geometric modeling, implicit functions are extensively used for surface modeling and rendering(97)(87), particularly for complex shapes and freeform surfaces. In physical simulations, implicit functions are employed to describe dynamic changes of object surfaces in fluid dynamics and elasticity(63)(40). In robotics, they help in path planning(21)(101) by representing obstacles and free spaces, thus aiding in obstacle avoidance. Moreover, implicit functions have demonstrated exceptional performance in other tasks including 3D reconstruction and semantic segmentation. Notably, the use of neural implicit representations, such as those employed in Convolutional Occupancy Networks(67), enables the incorporation of inductive biases like translational equivariance, facilitating the reconstruction of complex scenes from noisy point clouds and low-resolution voxel representations. The following are some common implicit functions, including Signed Distance Function (SDF)(66), Unsigned Distance Function (UDF)(15), and Occupancy Function(59):

The Signed Distance Function (SDF) represents the signed distance from any point in space to the target surface, with positive values for points outside the surface, negative values for points inside, and zero for points on the surface. Typically, applying SDF involves generating a regular grid in the target 3D space, then calculating the nearest distance from each grid point to the target surface and storing these distances in a 3D array or other data structures. Leveraging its ability to accurately represent the boundaries of complex geometries, providing a unified representation for closed surfaces, the representation in the SDF form is highly effective in accurately defining complex geometries, making it widely used in high-precision 3D reconstruction, collision detection, and shape analysis tasks. For instance, DeepSDF(66) leverages neural networks to represent continuous SDFs for classes of shapes, enabling high-quality shape representation, interpolation, and completion from partial and noisy 3D input data. Though traditional SDF methods are computationally intensive and require significant storage space, posing challenges for real-time updates in dynamically changing environments. However, these limitations have been alleviated by deep learning based approaches, which optimize storage and computation, allowing more efficient and scalable SDF representations.

The Unsigned Distance Function (UDF) measures the absolute distance from any point in space to the target surface, lacking the 'sign' information compared to SDF. While UDF cannot differentiate between the inside and outside of a surface, it is capable of reconstructing both closed and open surfaces when combined with other technologies like NeuralUDF, providing more flexibility for complex topologies than SDF, which necessitates the built-in nature to segmente shapes into inside and outside regions, thereby restricting its ability on open or highly complex surfaces. Recent advancements like NeuralUDF(55) introduces a differentiable indicator function to transform the UDF distance field into a volume density field, thereby enabling surface reconstruction in a manner similar to NeRF based approaches. Although this method effectively combines the advantages of UDF and volume density functions, addressing the inherent limitation of UDF in inferring occupancy statuses due to the lack of 'sign' information, it nonetheless suffers from optimization instability due to the inherent non-differentiability nature of UDF at zero-level sets, which poses challenges during model learning processes and hampers the accuracy of the reconstructed surfaces. Therefore, despite the computational efficiency and reduced storage requirements of UDF, its application necessitates a careful consideration of its advantages against its disadvantages, particularly regarding its suitability for tasks that require distinguishing between points inside and outside the surface. However, UDF is still advantageous in rapid reconstruction and basic segmentation tasks due to its computational efficiency.

The Occupancy Function is a binary indicator that specifies whether a point in space is occupied, with a value of 1 indicating occupancy and 0 indicating non-occupancy. This simple, intuitive representation is computationally efficient and ideal for large-scale 3D data processing, such as occupancy grid representation and robotic path planning. For example, Convolutional Occupancy Networks(67) extend traditional occupancy networks(59) by incorporating convolutional operations, enabling them to capture local and global spatial information effectively. This approach supports the detailed reconstruction of objects and large-scale scenes from noisy inputs, providing robust performance in real-world applications. Although the occupancy function itself lacks the ability for detailed geometric information, additional techniques such as that is proposed in Convolutional Occupancy Networks(67) take effect for capturing fine spatial details.

The next subsection briefly introduces the Convolutional Occupancy Network(67), which is also the basis on which the subsequent baseline of this paper is built.

2.1.1 Convolutional Occupancy Network

Convolutional Occupancy Network(67) presents an advanced framework for 3D reconstruction by combining the strengths of convolutional neural networks (CNNs) with implicit occupancy representations. This approach enables detailed and scalable 3D reconstructions of both objects and complex scenes, addressing the limitations of prior methods(59) that rely on simple fully-connected architectures.

Encoder:
The encoder architecture efficiently processes 3D input data such as point clouds or occupancy grids. For point cloud extraction, a fully-connected layer followed by ResNet blocks maps 3D point coordinates into a feature space. Features are locally pooled and concatenated before feeding into subsequent ResNet blocks, allowing effective aggregation of local information. A single 3D convolutional layer extracts voxel-wise features from occupancy grids. The features are then further processed by a U-Net, which handles plane or volume features, providing translational equivariance and integrating local and global information through down-sampling and up-sampling convolutions.

Additionally, the encoder is designed to handle different types of input representations: Point cloud encoder utilizes a shallow PointNet-like architecture with local pooling to efficiently encode point-wise features, which differs from traditional PointNet and enhancing the encoding of fine geometric details; While voxel encoder processes occupancy grids using a 3D convolutional layer, extracting voxel-wise features that encapsulate spatial information effectively.

Decoder:
The decoder employs a stack of fully-connected ResNet blocks to predict the occupancy probability of query points. By leveraging shallow architectures for memory efficiency, the decoder facilitates detailed and memory-efficient 3D reconstructions. The decoder architecture enables precise occupancy probability predictions using locally aggregated feature maps from the encoder. This combination of convolutional operations and fully-connected layers allows the model to reconstruct detailed 3D geometries accurately.

In short, the Convolutional Occupancy Network offers advantages in the field of 3D reconstruction over the following several points:

- Scalability: The convolutional nature allows the network to scale from single objects to entire scenes efficiently, making it suitable for both small and large-scale reconstructions.

- Translational Equivariance: By employing convolutions, the model incorporates inductive biases such as translational equivariance, enhancing its ability to generalize across different spatial configurations.
- Memory Efficiency: The use of shallow decoders and the reduction of parameter counts make the network more memory-efficient compared to traditional occupancy networks.
- Detailed Reconstruction: The Convolutional Occupancy Network excels in preserving fine geometric details, crucial for accurate 3D modeling, particularly from noisy or partial point clouds.

All in all, the Convolutional Occupancy Network represents a significant advancement in 3D reconstruction by integrating convolutional encoders and implicit occupancy decoders. This novel approach effectively handles complex geometries(object-level) and large-scale scenes(scene-level), providing a robust, scalable, and memory-efficient solution for 3D modeling tasks. The network's ability to preserve fine details and generalize across various datasets, as shown on real-world datasets like ScanNet(20) and Matterport3D(71), makes it a powerful tool for a wide range of applications in computer vision and beyond.

Based on the Convolution Occupancy Network, (58) proposed the Semantic Occupancy Network to leverage the occupancy function for simultaneous 3D reconstruction and semantic segmentation tasks, which forms the baseline for the subsequent research presented in this paper.

2.2 Semantic Scene Reconstruction

2.2.1 3D Reconstruction

Three-dimensional (3D) reconstruction is a vital field in computer vision, focusing on recreating the 3D geometry of objects or scenes from two-dimensional (2D) images or point clouds. This process is integral to various applications, including robotics, medical imaging, cultural heritage preservation, virtual reality and etc..

Traditional 3D reconstruction methods can be categorized into passive and active techniques. Passive methods like Structure from Motion (SfM)(74), which utilizes multiple 2D images from different viewpoints, detecting and matching key points across views to estimate camera parameters and reconstruct 3D structures. Stereo vision(75), another passive technique, uses images captured by two or more cameras from slightly different angles to compute depth information based on the disparity between the images.

Active techniques include laser scanning, structured light, and time-of-flight (ToF) cameras. Laser scanning(81) projects laser beams onto surfaces and measures the reflected light to obtain precise 3D coordinates. Structured light(2) involves projecting known patterns onto a scene, analyzing the deformation of these patterns to reconstruct 3D shapes. ToF cameras(38) estimate depth by measuring the time light takes to travel to and from the object.

Recent advancements incorporate Generative Adversarial Networks (GANs) into 3D reconstruction workflows to enhance various aspects. GANs(65) improve the resolution and accuracy of depth maps generated from stereo images, predict and fill occluded parts of objects, and generate realistic textures for 3D models, enhancing visual fidelity. Methods like 3D-GAN(91), Pix2Vox(95), and PC-GAN(32) illustrate the integration of GANs in generating 3D objects from 2D images or refining shapes from point cloud data.

Despite these advancements, several challenges remain in 3D reconstruction: Occlusions, where parts of the scene are hidden from certain viewpoints, and textureless surfaces are significant obstacles that complicate feature detection. Additionally, the computational complexity of high-resolution reconstructions and the dynamic nature of moving scenes pose further challenges.

Leveraging deep learning techniques, recent progress has significantly improved the accuracy and robustness of 3D reconstructions. The integration of multi-modal data, such as combining RGB images with depth sensors(94), has further enhanced reconstruction quality. The introduction of implicit functions, such as occupancy function(59), has further revolutionized this field by allowing the representation of 3D shapes without explicitly storing geometric data, leading to significant storage savings and efficient handling of complex geometries from limited input.

In conclusion, 3D reconstruction is a rapidly evolving field with substantial implications across various domains. Continuous development of new techniques and the integration of advanced machine learning models promise to overcome current limitations and drive further advancements in this essential area of computer vision.

2.2.2 Semantic Segmentation

Semantic segmentation is another pivotal task in computer vision, aiming to classify each pixel in an image or voxel in a 3D volume into a predefined category. This fine-grained understanding of scenes is critical in applications such as autonomous driving, medical imaging, and augmented reality, where precise localization and identification of objects are essential.

Traditional methods for semantic segmentation in 2D data involve techniques such as thresholding, edge detection, and region-based methods. Thresholding is simple but often ineffective in complex scenarios with varying illumination. Edge detection algorithms like the Canny(22) and Sobel(28) filters can delineate boundaries but lack semantic context. Region-based methods such as Region Growing(1) and Watershed(76) group pixels based on similarity, but they struggle with texture variations and noisy data.

The advent of deep learning has significantly enhanced the performance of semantic segmentation. Fully Convolutional Networks (FCNs)(54) marked a breakthrough by replacing fully connected layers with convolutional layers, enabling dense pixel-wise predictions. 3D U-Net(17), particularly effective in biomedical imaging, employs an encoder-decoder architecture with skip connections that preserve spatial information. DeepLab(11), utilizing atrous convolutions and Conditional Random Fields (CRFs), excels in capturing multi-scale contextual information and refining object boundaries.

For 3D data, semantic segmentation methods extend these concepts to volumetric data. Voxel-based methods often use 3D Convolutional Neural Networks (3D CNNs)(30) to process volumetric data, enabling the segmentation of 3D medical images and point clouds. However, these methods are computationally intensive and memory-demanding. Multi-view approaches project 3D data into multiple 2D views, leveraging established 2D segmentation techniques but facing challenges in merging results from different views, often accompanied by potential semantic boundary inconsistency issues.

Implicit function representations offer significant advantages for semantic segmentation in both 2D and 3D data. These functions, such as Signed Distance Functions (SDFs)(66) and occupancy fields(59), provide continuous and memory-efficient representations, allowing for detailed and smooth surface reconstructions. They adapt seamlessly to various resolutions and scales, enhancing flexibility and efficiency in processing.

Despite these advancements, several challenges persist: Class imbalance, where some categories are underrepresented in training datasets, can lead to poor model performance for those classes. Achieving precise boundaries between objects remains difficult, especially in complex scenes with occlusions and overlapping objects. The computational demand for processing high-resolution images or volumetric data is substantial, necessitating efficient algorithms and powerful hardware.

Current techniques exhibit a range of strengths and weaknesses. FCNs(54) are straightforward and effective but may miss fine details. 3D U-Net(17) performs exceptionally well in specific domains like medical imaging but can be computationally intensive. DeepLab(11) achieves high accuracy and precise boundary delineation but requires careful hyperparameter tuning. In 3D segmentation, voxel-based methods provide detailed volumetric segmentation but face significant computational and memory constraints(50). Implicit function-based methods offer efficiency and flexibility but integrating them into existing deep learning frameworks can be complex in some cases.

All in all, semantic segmentation is a dynamically evolving field, driven by advancements in deep learning and computational techniques. The integration of implicit functions presents new opportunities for efficient and precise segmentation of both 2D and 3D data. However, challenges such as class imbalance, boundary precision, and computational demands necessitate ongoing research and innovation. Future work is expected to focus on such aspects including enhancing model efficiency, improving boundary precision, and broadening the applicability of semantic segmentation across diverse domains.

2.2.2.1 3D U-Net

The application of convolutional neural networks (CNNs) to biomedical image segmentation has demonstrated remarkable success, particularly with the introduction of the U-Net(72) architecture for 2D images. However, many tasks, especially in biomedical imaging and large-scale outdoor environments, require volumetric segmentation. Extending 2D methods to these domains is non-trivial due to the inherent complexity and higher-dimensional nature of the data. To address these challenges, the 3D U-Net(17) architecture was developed, leveraging 3D convolutions to perform dense volumetric segmentation from sparse annotations.

The 3D U-Net extends the 2D U-Net by replacing 2D operations with their 3D counterparts. This architecture(see figure 2.1) comprises a contracting path (encoder) and an expanding path (decoder). The encoder captures context through repeated application of 3x3x3 convolutions, followed by rectified linear unit (ReLU) activations and 2x2x2 max pooling operations. This process progressively reduces spatial dimensions while increasing the number of feature channels. Conversely, the decoder path mirrors the encoder, utilizing 3D up-convolutions to progressively restore spatial dimensions and refine the segmentation map. Skip connections between corresponding layers of the encoder and decoder paths facilitate the transfer of high-resolution features, which are critical for precise localization.

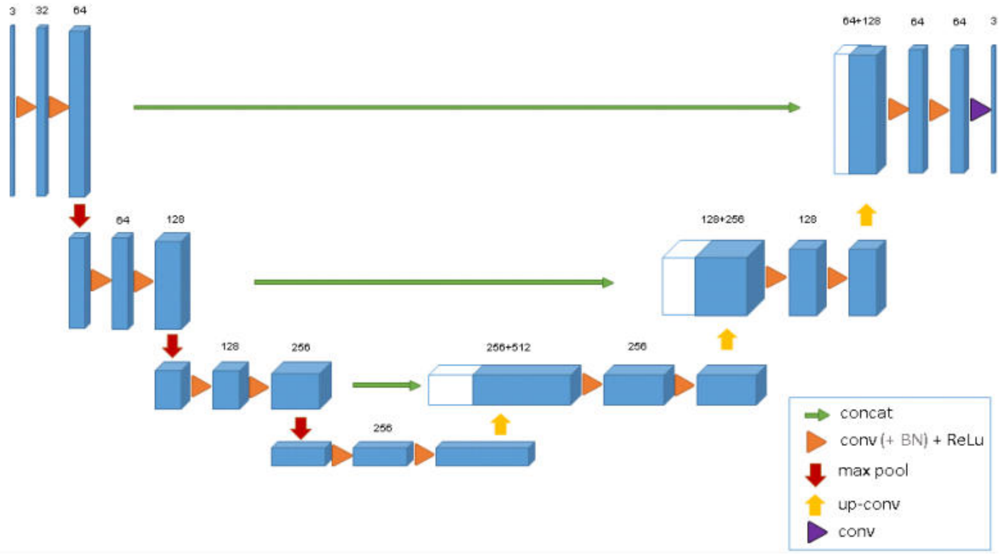


Figure 2.1: Architecture of 3D U-Net(17)

Integral components of the 3D U-Net architecture include 3D convolutions, 3D max pooling, and 3D up-convolutions. The network also incorporates batch normalization layers, which standardize the inputs to each layer, accelerating training convergence and improving model performance. To handle sparse annotations, a weighted softmax loss function is employed by assigning zero weights to unlabeled pixels, enabling effective learning from partially annotated volumes.

One primary advantage of the 3D U-Net is its ability to perform accurate and efficient volumetric segmentation from sparse annotations, significantly reducing the annotation burden in biomedical applications. Its design captures both local and global context, facilitating precise segmentation even in complex structures. Furthermore, the use of batch normalization enhances training stability and model generalization. However, the high computational and memory demands of processing 3D volumes pose significant challenges. Training 3D U-Net models requires substantial computational resources and time, which could be a potential limiting factor in some cases.

The 3D U-Net is particularly well-suited for semantic segmentation tasks in various fields including biomedical domain and large-scale outdoor environments due to its ability to handle volumetric data and learn from sparse annotations. In medical imaging, obtaining fully annotated datasets is often impractical, making the 3D U-Net's capabilities crucial. The architecture's effectiveness has been demonstrated in applications such as brain tumor segmentation, liver and kidney segmentation, and organ and tissue delineation tasks. Additionally, in outdoor scenes, 3D U-Net can segment complex environments, aiding in autonomous driving and urban planning by generating high-resolution segmentation maps that improve object recognition and spatial understanding.

Several notable models and variations have been developed based on the 3D U-Net architecture to enhance its performance for specific tasks. For instance, the V-Net(61) incorporates residual connections and a Dice loss function, which is particularly effective for dealing with class imbalance common in medical imaging. Additionally, the Attention U-Net(64) integrates attention mechanisms to focus on relevant features and suppress irrelevant background noise, further improving segmentation accuracy. Extensions such as the 3D U-Net++(107) introduce dense skip connections and deep supervision to refine the segmentation output.

In summary, the 3D U-Net architecture represents a significant advancement in volumetric image or voxel-like segmentation, particularly for biomedical and large-scale outdoor applications. By extending the 2D U-Net to handle 3D data, the 3D U-Net effectively addresses the challenges of volumetric segmentation, leveraging sparse annotations to produce dense and accurate segmentation maps. Despite its computational challenges, the 3D U-Net's ability to capture detailed context and generate high-resolution segmentations makes it an invaluable tool in advancing spatial analysis. Future research is likely to focus on optimizing computational efficiency and further enhancing segmentation accuracy, ensuring the continued applicability and impact of the 3D U-Net in diverse domains.

2.2.2.2 SwinTransformer

The Swin Transformer(52), a hierarchical vision Transformer, effectively addresses the limitations of traditional Transformer models in computer vision, particularly for tasks involving large-scale visual entities and high-resolution images. Unlike conventional Transformers, which struggle with the varying scales of visual elements and exhibit quadratic computational complexity, the Swin Transformer employs a hierarchical architecture with shifted windows to mitigate these issues. This design makes it suitable for a diverse range of vision tasks, including image classification, object detection, semantic segmentation, pose estimation and etc..

The architecture(see figure 2.2) of the Swin Transformer includes several key components. Initially, an input image is divided into non-overlapping patches, which are treated as tokens and embedded into feature vectors using a linear embedding layer. The architecture then utilizes a series of Swin Transformer blocks, each comprising multi-head self-attention modules with shifted windows and multi-layer perceptrons (MLPs). The shifted window approach confines self-attention computation within non-overlapping windows, significantly reducing computational complexity. The hierarchical design enables the network to process features at multiple scales, effectively capturing both local and global context.

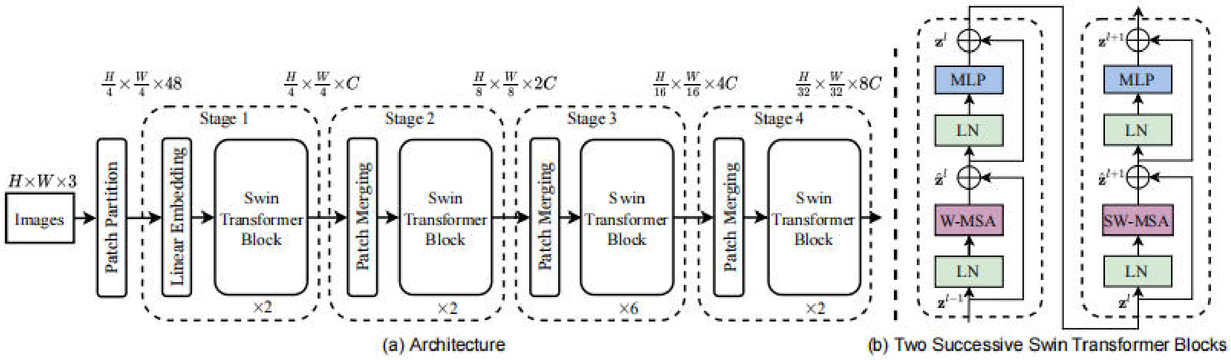


Figure 2.2: Architecture of Swin Transformer, where LN denotes a linear layer, W-MSA stands for windowed multi-head self-attention layer, and SW-MSA extends W-MSA by incorporating a shifting operation, MLP represents a multi-layer perceptron.(52)

One of the primary advantages of the Swin Transformer is its linear computational complexity concerning image size, achieved by restricting self-attention to local windows. This capability allows the model to handle high-resolution images and dense prediction tasks efficiently. Additionally, the shifted window mechanism introduces cross-window connections, enhancing the model's ability to capture long-range dependencies and improving overall performance. The incorporation of relative position bias further boosts the model's effectiveness by maintaining spatial relationships between patches.

The Swin Transformer is particularly well-suited for semantic segmentation tasks. Its hierarchical feature maps align seamlessly with existing dense prediction techniques such as feature pyramid networks (FPN)(46) and 3D U-Net(17), making it a versatile backbone for various vision tasks. In semantic segmentation, the Swin Transformer surpasses previous state-of-the-art models, achieving notable improvements with metrics such as mean Intersection over Union (mIoU). For example, it achieves 53.5 mIoU on the ADE20K(106) validation set, outperforming previous models by a substantial margin.

Several models have incorporated the Swin Transformer to enhance their performance in specific tasks. For instance, the Swin Transformer has been integrated into object detection frameworks like Cascade Mask R-CNN(9), ATSS(5), and RepPoints V2(12), achieving higher accuracy and efficiency compared to traditional CNN-based backbones. Its application in 3D reconstruction tasks benefits from its ability to model fine-grained details and capture complex spatial relationships, crucial for generating accurate 3D representations of scenes.

In large outdoor scenes, the Swin Transformer excels in semantic segmentation and 3D reconstruction by effectively handling high-resolution inputs and modeling the intricate details and variability of outdoor environments(52). Its ability to generate hierarchical feature maps and maintain linear computational complexity ensures that it can process large-scale data efficiently, making it an ideal choice for these demanding tasks.

In conclusion, the Swin Transformer represents a significant advancement in vision Transformers, providing a robust and efficient solution for a wide range of computer vision tasks. Its hierarchical architecture, shifted window approach, and linear computational complexity make it particularly effective for semantic segmentation and 3D reconstruction. Future research is likely to explore further optimizations and applications of the Swin Transformer, cementing its role as a versatile backbone in computer vision.

2.3 Point Feature Encoding

2.3.1 PointNet Based Variant

PointNet(68) is a neural network architecture designed specifically for processing point clouds. Based on PointNet, several variants have been developed. In this work, we employ one of these variants. Similarly, the variant converts input points into a local coordinate system based on a predefined voxel size, allowing it to handle a variable number of point cloud inputs, but necessitating precomputed local pooling indices. Then it integrates multiple ResNet modules and subsequently combines a 3D U-Net structure to enhance feature refinement. The basic structure of the PointNet-based variant is illustrated in figure 2.3:

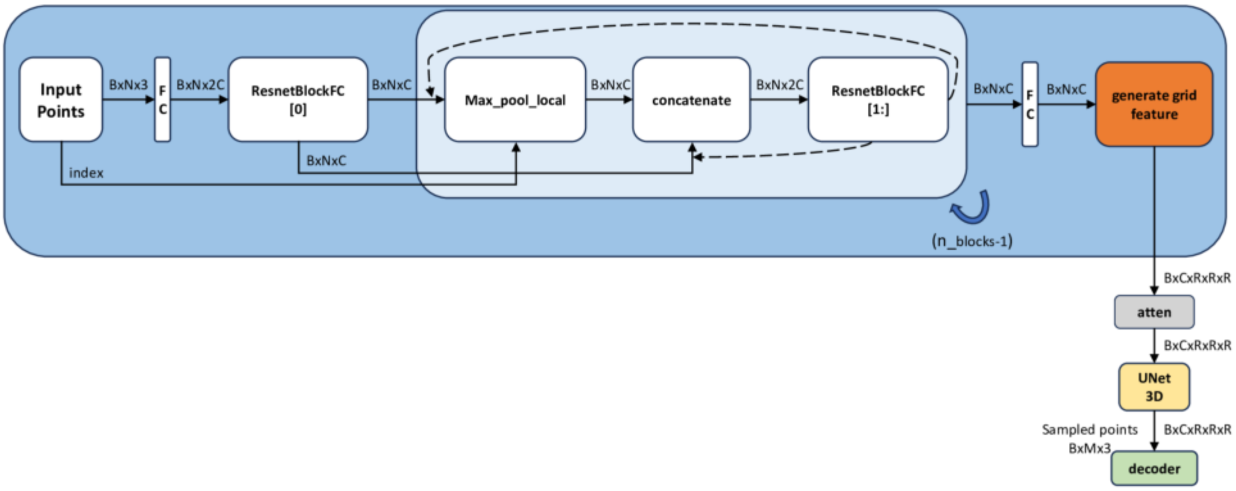


Figure 2.3: Basic structure of PointNet based variant.

Initially, the positional data of the input points are processed through a linear layer to increase the channel dimensions for subsequent feature encoding. The first ResNet module adjusts the channel dimensions, followed by a local max pooling operation before each new ResNet module is introduced. The features obtained are concatenated with the output of the previous ResNet module and then input into a new ResNet module, implementing a specified number of recycles. Finally, after adjusting the channel dimensions, grid features of the specified size are generated and input into 3D U-Net for further feature extraction based on the context.

2.3.2 PointNet++

Compared to PointNet, PointNet++(69) employs a hierarchical feature extraction approach, with each feature extraction combination referred to as a set abstraction layer. Each set abstraction layer consists of three components: sampling, grouping, and feature extraction.

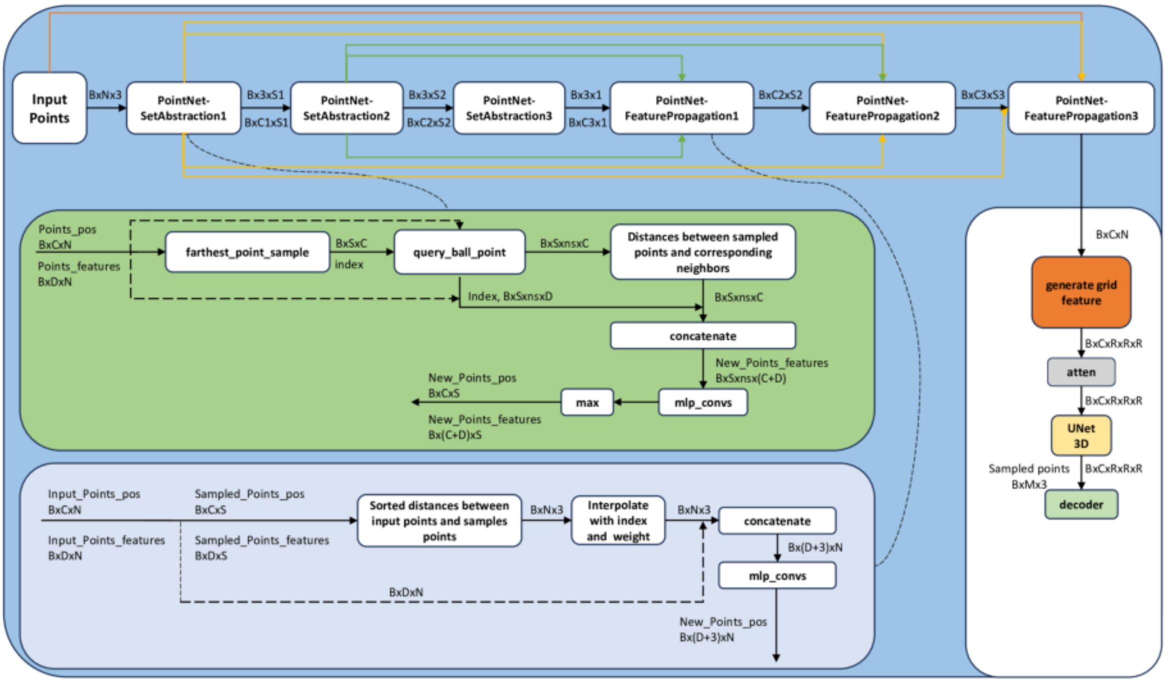


Figure 2.4: Basic structure of PointNet++.

In the sampling step, the goal is to extract important central points from the dense point cloud. This is achieved using the farthest point sampling (FPS) method to obtain the desired number of points for each group. These points do not necessarily contain semantic information. Following sampling, grouping is performed. For each central point obtained from the sampling step, k-nearest neighbors within a spherical region are identified to form a group. In the feature extraction step, the k-nearest neighbors of each sampled point are processed through multiple convolutional and pooling operations. The resulting features are treated as the features of the sampled point, analogous to a simplified version of PointNet feature extraction for each sampled point with respect to its neighborhood.

The above-mentioned three steps constitute one set abstraction layer. PointNet++ typically employs three such layers, with each successive layer reducing the number of central points while increasing the amount of information contained in each point.

Additionally, PointNet++ addresses the issue of non-uniform sampling density, where fixed-range selection of a fixed number of nearest neighbors might not adequately represent features for each sampled point. Two solutions are proposed: Multi-Scale Grouping (MSG) and Multi-Resolution Grouping (MRG).

Multi-Scale Grouping (MSG): Each grouping layer determines groups using multiple scales (different neighborhood radii). Features are extracted using a simplified version of PointNet and concatenated to form new features.

Multi-Resolution Grouping (MRG): Features from different layers are combined. For example, one branch might use two consecutive set abstraction layers with decreasing sampling sizes and different radii, while another branch uses one set abstraction layer with the same sampling size as the first set abstraction layer in branch 1 but different radii. Features from both branches are weighted and concatenated, with weights adjusted based on point cloud density. If point cloud density is low, the patch information learned in the subsequent set abstraction layers of the first branch might have low reliability, and the weight of the second branch could be increased in this case.

Finally, the features obtained are propagated to each input point through inverse distance weighted interpolation between the low-resolution and high-resolution point sets, effectively passing the learned features from each set abstraction layer to every input point.

2.4 Attention Mechanism

The attention mechanism, as proposed by(82), maps the query and a set of key-value pairs to the output through the "scaled dot-product attention" process. The core principle involves calculating the similarity between the query and all keys, subsequently using these similarities to perform a weighted summation of the corresponding values. The weights are determined by the compatibility function between the query and the corresponding key.

The detailed process is as follows:
1) Compute the dot products of the queries and keys;
2) Scale the dot product by dividing the square root of d_k;
3) Apply the softmax function to obtain the weights;
4) Use these weights to perform a weighted sum of the values, producing the final output;

The formulas are as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.1}$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2.2}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

To capture information from different representation subspaces, the Transformer(82) architecture introduces a multi-head self-attention mechanism. Each attention head independently executes the aforementioned computations, concatenates the results, and performs a final linear transformation to produce the output.

2.4.1 SwinVFTR

To enhance the performance of semantic scene reconstruction, integrating attention mechanisms into the 3D U-Net component is a viable option. This can improve the voxel-like feature extraction block. While exploring methods to combine SwinTransformer(52) with 3D U-Net(17), we discovered the SwinVFTR network(37), which effectively integrates these elements. Although originally developed for brain tumor segmentation, its architecture is well-suited for semantic scene reconstruction tasks. The following (see figure 2.5) illustrates its structure:

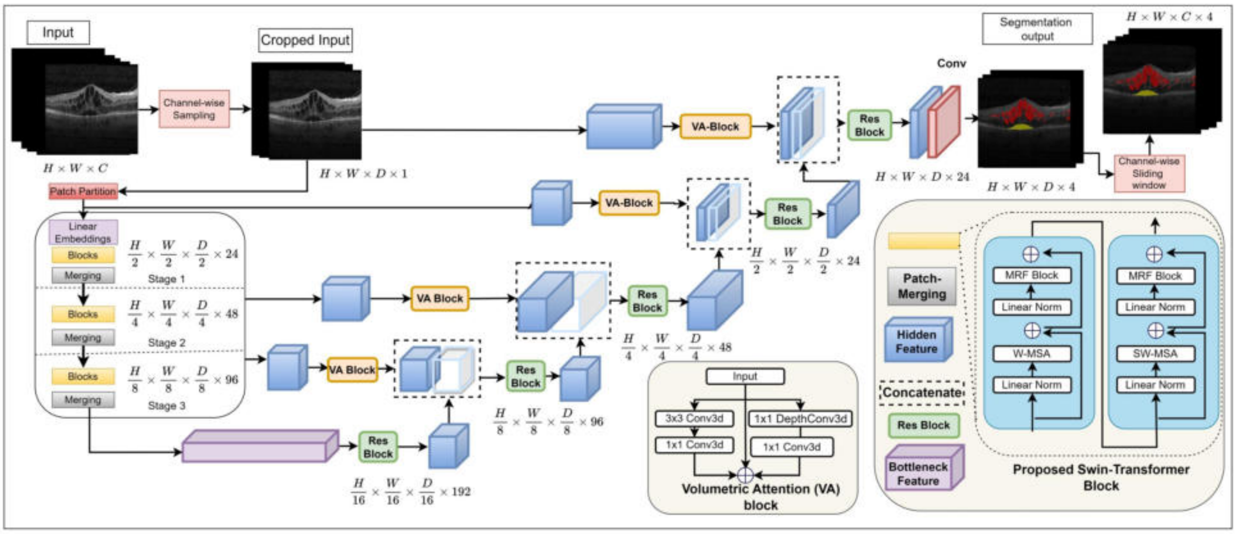

Input
Cropped Input
Channel-wise Sampling
$H \times W \times C$
$H \times W \times D \times 1$
Patch Partition
Linear Embeddings
Blocks
Merging
$\frac{H}{2} \times \frac{W}{2} \times \frac{D}{2} \times 24$
Stage 1
Blocks
Merging
$\frac{H}{4} \times \frac{W}{4} \times \frac{D}{4} \times 48$
Stage 2
Blocks
Merging
$\frac{H}{8} \times \frac{W}{8} \times \frac{D}{8} \times 96$
Stage 3
VA-Block
VA-Block
VA Block
VA Block
Res Block
Res Block
Res Block
Res Block
Res Block
$\frac{H}{16} \times \frac{W}{16} \times \frac{D}{16} \times 192$
$\frac{H}{8} \times \frac{W}{8} \times \frac{D}{8} \times 96$
$\frac{H}{4} \times \frac{W}{4} \times \frac{D}{4} \times 48$
$\frac{H}{2} \times \frac{W}{2} \times \frac{D}{2} \times 24$
Conv
$H \times W \times D \times 24$
$H \times W \times D \times 4$
Segmentation output
$H \times W \times C \times 4$
Channel-wise Sliding window
Input
3x3 Conv3d
1x1 DepthConv3d
1x1 Conv3d
1x1 Conv3d
Volumetric Attention (VA) block
Patch-Merging
Hidden Feature
Concatenate
Res Block
Bottleneck Feature
MRF Block
Linear Norm
W-MSA
Linear Norm
MRF Block
Linear Norm
SW-MSA
Linear Norm
Proposed Swin-Transformer Block

Figure 2.5: Basic structure of SwinVFTR.(37)

The Swin Volumetric Feature-learning Transformer (SwinVFTR) architecture integrates the Swin-Transformer with a 3D U-Net structure to address the challenges of 3D image segmentation. This hybrid approach leverages the strengths of both transformer-based networks and 3D U-Net architectures to achieve superior performance in segmentation tasks.

The SwinVFTR network architecture begins with a channel-wise volumetric sampling technique that preprocesses input volumes to ensure compatibility with varying depths. This sampling retains spatial information while adjusting the depth dimension, ensuring that the model can handle inputs with different depths effectively.

The encoder of the SwinVFTR utilizes a novel Swin-Transformer block. This block integrates shifted windows multi-head self-attention (SW-MSA) to capture hierarchical local features efficiently. The SW-MSA mechanism divides the input volume into smaller windows, applying attention within each window and shifting the windows to capture more global context. This approach balances the computational complexity while maintaining high feature representation quality. Additionally, the Swin-Transformer block in the SwinVFTR replaces the traditional multi-layer perceptron (MLP) with a Multi-receptive field (MRF) residual block. This MRF block consists of parallel convolutions: standard, depth-wise, and dilated convolutions. This design captures features at multiple scales and enhances the block's ability to model complex spatial patterns within the volumetric data.

The encoded features are progressively downsampled using patch-merging layers, which combine features from neighboring patches and reduce their spatial dimensions, effectively capturing multi-scale information. Each stage in the encoder consists of two Swin-Transformer blocks followed by a patch-merging layer, ensuring the features are hierarchically aggregated.

For the decoder, SwinVFTR employs a symmetric design, using UnetrUp blocks for upsampling the feature maps back to the original resolution. These blocks use transposed convolutions to increase the spatial dimensions and incorporate residual connections to maintain feature integrity. The decoder stages also use volumetric concatenation layers that merge features from corresponding encoder stages via skip connections, ensuring that detailed spatial information is preserved.

A key innovation in SwinVFTR is the introduction of Volumetric Attention (VA) blocks within the skip connections. These blocks enhance the feature maps by applying spatial and channel-wise attention mechanisms. The spatial attention is achieved using standard and point-wise convolutions, while the channel-wise attention is handled by depth-wise convolutions, ensuring that both spatial and depth information is effectively emphasized.

The final segmentation map is produced through a 1x1x1 convolutional layer followed by a softmax activation function, which generates the probabilistic segmentation of the input volume.

The SwinVFTR architecture's design advantages lie in its ability to capture fine-grained details through multi-receptive field processing, maintain high-resolution spatial information via efficient skip connections, and effectively model hierarchical features through the shifted windows attention mechanism. This combination makes SwinVFTR particularly suited for tasks requiring precise 3D reconstruction and semantic segmentation, such as the analysis of complex structures or volumetric data in semantic scene reconstruction, which aligns with our objectives of achieving high precision in simultaneous dual tasks.

2.5 Class Imbalance Problem

Class imbalance, a prevalent issue in computer vision, arises when certain classes are underrepresented in a dataset. This imbalance often leads to biased models that perform well on frequent classes but poorly on rare ones. This issue is particularly prominent in applications such as medical imaging, autonomous driving, and large-scale outdoor scene segmentation, where the distribution of objects or features is inherently uneven.

In medical imaging, for instance, normal tissues might vastly outnumber pathological tissues, leading to models that fail to detect diseases accurately. Similarly, in autonomous driving, rare events like pedestrian crossings occur less frequently compared to other static objects like roads and buildings, causing the model to overlook critical yet infrequent occurrences.

To address class imbalance, various strategies have been developed, broadly categorized into data-level, algorithm-level, and hybrid approaches. Data-level approaches include oversampling, which involves duplicating instances of minority classes to balance the dataset(96), although this can lead to overfitting. Undersampling reduces instances of majority classes to achieve balance but risks losing valuable information. Synthetic data generation techniques like SMOTE(10)(Synthetic Minority Over-sampling Technique) create synthetic instances for minority classes, enhancing diversity but potentially introducing noise if not carefully implemented.

Algorithm-level approaches consist of cost-sensitive learning(26), which assigns higher misclassification costs to minority classes to encourage the model to prioritize these classes. This method can be highly effective but requires careful tuning of cost parameters. Another method is class reweighting, which adjusts the loss function to give more importance to minority classes. This approach integrates seamlessly with existing models but may necessitate extensive experimentation to determine optimal weights.

Hybrid approaches combine data-level and algorithm-level techniques to provide robust solutions. For example, combining oversampling with class reweighting leverages the strengths of both methods (31). Each approach has its advantages and disadvantages: data-level methods like oversampling and synthetic data generation can increase the risk of overfitting and noise, respectively, while undersampling may lead to the loss of valuable information. Algorithm-level methods such as cost-sensitive learning and class reweighting require meticulous parameter tuning and might increase computational complexity.

In large-scale outdoor scene segmentation and 3D reconstruction, class imbalance poses significant challenges like the critical yet infrequent occurrences above mentioned. Class reweighting has several benefits for these applications. It allows the model to focus on underrepresented classes without altering the original data distribution. This approach leads to improved detection and segmentation of rare but critical objects, enhancing the overall performance and robustness of the model. Additionally, class reweighting can be easily integrated into various deep learning frameworks, making it a flexible and scalable solution for addressing class imbalance in large-scale vision tasks.

In conclusion, addressing class imbalance is crucial for developing accurate and reliable computer vision models. While various methods exist, each with its trade-offs, combining these approaches can provide robust solutions tailored to specific applications. Future research should continue to explore and refine these techniques, particularly in the context of large-scale and complex environments, to ensure comprehensive and balanced model performance.

3 Related Work

This chapter provides a detailed overview of the evolution of methodologies employed in the realm of 3D reconstruction and semantic segmentation. Section 3.1 delves into the exploration of deep implicit function based surface reconstruction. Subsequently, Section 3.2 discusses the recent advancements in the semantic scene reconstruction pipeline. Following this, in Section 3.3 and 3.4, a general view of point feature encoding and the class imbalance problem inherrent in semantic segmentation are separately carried out, which are the main area of concentration that ties into the overarching theme of this thesis.

3.1 Deep Implicit Function Based Surface Reconstruction

In the field of 3D surface reconstruction, significant advancements have been made over the past decades, evolving from the initial restoration of physical shapes of individual objects with defective scanned digital representations to encompassing a wide variety of indoor and outdoor objects and scenes, including the evolution of objects from static primitives to dynamic ones and non-explicit geometries. Traditional methods, including active methods such as optical laser-based range scanners, structured light scanners, LiDAR scanners, and passiv methods such as multi-view stereo, primarily leverage various devices to rapidly execute engineering and prototyping tasks, benefiting computer-aided design and computer graphics. However, these hardware-based data acquisition methods pose different challenges for surface reconstruction tasks due to the varying nature of the data they produce. (4) provides a comprehensive classification of various surface reconstruction techniques by considering the types of data priors, the deficiencies inherent in the acquired data, and the resulting reconstruction outcomes, which offers an overarching perspective on the advancements and methodologies within the field of surface reconstruction, highlighting the strengths and limitations of different approaches based on the nature of the input data and the desired quality of the reconstructed surfaces.

Despite the advancements achieved by traditional methods, they still inherently possess fundamental limitations. In recent years, the latest developments in deep learning have endowed data-driven approaches with the ability to optimize the inherent defects arising from data acquired from various devices, thereby becoming the mainstream in research. With the burgeoning development of implicit functions, researchers have integrated deep learning techniques with these functions, resulting in numerous outstanding methods across various domains, including surface reconstruction:

Deep Implicit Moving Least-Squares Functions(49) integrates the flexibility of point sets with the high-quality representation of implicit surfaces, using an octree-based scaffold to adaptively generate MLS points, improving computational efficiency and reconstruction accuracy. Structured Local Deep Implicit Functions(29) decomposes 3D shapes into a set of structured local implicit functions, each associated with a latent vector that captures fine geometric details. The coarse structure is defined by Gaussian functions, while the fine details are represented by local deep implicit functions. This approach enhances the accuracy of surface reconstruction while improves generalization to unseen shapes, and reduces the required network parameters, thus achieving a more efficient and detailed reconstruction process. In contrast to (29), Multiresolution Deep Implicit Functions(13) introduces a multiresolution framework that leverages deep implicit functions in a different manner to capture shape features at various levels of detail, both globally and locally. Building upon (29) , LP-DIF(86) also segments 3D shapes into local regions. However, it further enhances this concept by clustering regions with similar geometric patterns and employing distinct, pattern-specific decoders for each cluster. Additionally, a region re-weighting module is integrated to address data imbalance issues, thereby improving the reconstruction of details in sparsely observed regions. (44) represents another advancement based on the methodology in (29). Unlike (29), which utilizes fixed local regions, (44) introduces dynamic code clouds and a novel loss function to guide the distribution of local codes towards regions with high geometric complexity, which dynamically optimizes the effective positions of local latent codes, enhancing representational capacity and improving efficiency. (53) employs hierarchical feature maps and permutohedral lattices to efficiently encode and query local implicit functions. Compared to the methods based on (29), this approach also aims to enhance the detail and accuracy of 3D reconstruction using local implicit functions, while offering superior scalability.

Additional notable approaches in conjunction with implicit functions to achieve 3D reconstruction tasks include:

3DIAS(98) uses multivariate polynomials to design implicit algebraic surfaces for predefined shapes, enabling detailed reconstruction of complex geometries with fewer parameters. (16) introduces a two-stage network architecture combining the recently popular Transformers(82), where the first stage employs a 3D sparse stacked hourglass network for initial voxel generation and denoising, and the second stage uses Transformers for voxel re-localization, converting discrete voxels into precise 3D points. PointConv(93) addresses the challenge of non-uniform sampling by treating convolution kernels as continuous functions and reweighting them using density scales. MV-DeepSDF(51) transforms the implicit space shape estimation problem into an element-to-set feature extraction problem, utilizing global features and latent codes from multi-sweep point clouds combined with SDF for 3D vehicle reconstruction. (104) introduces the concept of deep implicit templates, decomposing a conditional deep implicit function into a template implicit function representing the mean shape and a conditional spatial warping function that deforms this template to match specific instances, allowing dense correspondences among different instances while maintaining compactness and efficiency.

While these methods each have their own advantages, they also exhibit potential drawbacks. For instance, (49) involves the complexity of managing the octree structure and the computational overhead of generating and evaluating MLS points for large-scale or detailed models. (29) entails computational costs in decomposing shapes into local elements and managing the alignment of local implicit functions. (13) experiences increased errors in unobserved regions. (77) demands higher memory for each voxel and introduces complexity in managing gradient updates, potentially affecting large-scale reconstruction performance. (98) struggles with handling entirely new, irregular geometries not covered by predefined shapes. (44) presents significant challenges in optimization complexity. (53) incurs computational overhead during the feature extraction and querying processes and etc.. In our dual task of semantic scene reconstruction, the accuracy of surface reconstruction is a prerequisite for segmentation. Therefore, our pipeline in the geometric component favors a method that offers high accuracy in representation while maintaining low memory consumption.

3.2 Semantic Scene Reconstruction

Semantic scene reconstruction aims to jointly estimate the geometric and semantic information of a 3D scene from sparse surface measurements(58). Most methods in the literatures achieve this task sequentially and present the results in the form of a voxel grid. Since semantic scene reconstruction fundamentally consists of two subtasks — 3D reconstruction and semantic segmentation — the former provides precise geometric information, which is a crucial prerequisite for the latter. Following the discussion in Section 3.1, which provides a clear understanding of recent developments in 3D surface reconstruction using implicit functions, we now explore the integration of 3D surface reconstruction and semantic segmentation.

Effectively leveraging the potential synergy between geometric and semantic information in scenes has long been a challenging aspect of the semantic scene reconstruction task. Geometric information can serve as prior knowledge to determine the semantic nature of entities, while semantic labels can constrain the characteristics of associated geometries. To overcome these limitations, recent models have made several attempts:

These methods can be broadly categorized based on different input modalities, including monocular (105), binocular(62), multi-view stereo(83) (92), and point clouds(14):

(83) combines a hash-based fusion approach for 3D reconstruction with volumetric mean-field inference for semantic segmentation. By utilizing stereo image pairs to generate depth information, a dense semantic 3D map is incrementally built and fused into a common 3D map. A Conditional Random Field (CRF) framework is employed for semantic labeling, with unary potentials derived from a random forest classifier and pairwise potentials ensuring smoothness by leveraging 3D features such as appearance, depth, and surface normals. This method efficiently manages memory using a hash-table-driven space allocation and effectively integrates moving objects into the reconstruction process. However, its performance is highly dependent on the quality of depth estimation, which is always noisy in outdoor environments. Similar to(83), (92) also uses stereo images to incrementally obtain depth information for building a 3D map but places greater emphasis on the integrity of occluded regions.

(14) utilizes sparse convolutions to integrate sparse tensor representations of point clouds, predicting occupancy and class labels for each voxel to semantically label the scene after incremental 3D reconstruction.

(105) employs a variational autoencoder(VAE) framework to encode geometric and semantic information from monocular images into a compact latent space. By optimizing low-dimensional codes associated with overlapping images, it ensures spatial consistency in the fused label maps, which excels at preserving spatial correlations across multiple views, generating consistent and smooth semantic labels, thus overcoming the noise issues associated with independent label estimation. (36) introduces a novel approach to reconstruct semantically labeled 3D scenes using only 2D image annotations. By leveraging differentiable rendering, it links 2D observations with the unobserved 3D space, using RGB images and 2D semantics for supervision. Although this method achieves state-of-the-art performance in semantic scene completion without relying on costly 3D annotations, it may face challenges in handling diverse real-world scenarios and ensuring the accuracy of 3D reconstruction predictions.

(62)integrates instance segmentation, feature matching, and point-set registration to enable real-time 3D scene perception and understanding for robots. By using YOLOv8 for object segmentation in RGB images and mapping 2D correspondences between consecutive frames into 3D correspondences via depth maps, kernel density estimation(KDE) aligns these correspondences for robust point cloud registration. However, this method introduces complexity in segmentation and feature matching processes and increases sensitivity to depth perception errors, particularly with transparent or shiny objects.

Building on the foundation of a geometric component that is able to offer high precision for subsequent accurate segmentation and low memory consumption requirement that makes the model avoid the hamper of potentially limited computational resources when attempting to enhance performance for segmentation tasks, influenced by S3CNet(14), we transfer our attention on the occupancy function, which has the potential to achieve the dual task without introducing techniques that incur additional high memory consumption.

3.3 Point Feature Encoding

Point feature encoding has become a pivotal technique in 3D computer vision, enabling effective processing and understanding of point cloud data. The primary goal of point feature encoding is to transform the irregular, unordered structure of point clouds into a format suitable for deep learning algorithms. The following highlights the significant advancements in point feature encoding, starting from PointNet and extending to its various derivatives and other contemporary methods:

Introduced by (68), PointNet was a groundbreaking method that directly processes unordered point clouds without the conventional need for voxelization or rendering into images. PointNet utilizes shared Multi-Layer Perceptrons (MLPs) and a max pooling layer to aggregate global features, effectively maintaining invariance to input permutations. This approach demonstrates strong performance in 3D object classification, part segmentation, and scene semantic parsing. Based on PointNet, PointNet++(69) was extended, incorporating a hierarchical structure to capture local geometric features at multiple scales. PointNet++ recursively applies PointNet on nested partitions of the pointset, significantly enhancing its capability to handle varying point densities and complex scenes.

Then further contributions are made on dynamic feature aggregation. Proposed by (90), DGCNN dynamically constructes a local graph for each point and applies edge convolutions to capture local geometric relationships, which improves feature learning by adapting to the dynamic nature of point cloud structures, leading to enhanced robustness and accuracy.

With the emergence of transformer(82), subsequent researchers attempted to embed this module into point cloud processing to further improve the accuracy of point features. (103) developed the Point Transformer, which integrates self-attention mechanisms to encode point cloud features, which allows the model to capture long-range dependencies and provides robustness to variations in point cloud density and distribution. Building on the requirement of local feature aggregation methods, (34) introduced FPTransformer, which utilizes local position encoding to enhance the awareness of each point within its local shape context, improving the robustness and accuracy of feature encoding, especially in capturing fine-grained details.

Other feature encoding techniques includ iterative and point-wise aggregation, such as PCRNet(73), which leverages PointNet for point cloud registration. PCRNet iteratively refines the alignment between point clouds using a loss function based on Earth Mover's Distance(EMD), enhancing robustness and accuracy. FPConv(48) revisites local point aggregations, including MLPs, point convolutions, and transformers, to derive a general formulation for local feature aggregation. FPConv introduces point convolutions with learned weights from local point coordinates, which dynamically adjustes point-wise features based on local geometric information.

The advancements in point feature encoding, starting from PointNet and extending to its variants and other contemporary methods, have significantly enhanced the performance of point cloud processing in various 3D computer vision tasks. Each method has contributed unique approaches to overcoming challenges such as handling unordered pointsets, capturing local and global features, and ensuring robustness to input variations. These innovations continue to drive the field forward, addressing limitations and exploring new possibilities for 3D data understanding.

However, when applying these techniques to representing large outdoor scenes with sparse point clouds from surface measurenments as inputs, certain considerations must be made. PointNet and its derivatives excel at handling unordered point sets and capturing global features efficiently. However, their reliance on global pooling can cause loss of fine local details, which are crucial in complex outdoor environments. Methods like PointNet++ and DGCNN improve local feature capture through hierarchical and dynamic graph-based structures but can be computationally intensive and may struggle with very sparse or unevenly distributed point clouds. The ideal point feature encoding method for large outdoor scenes should balance computational efficiency with the ability to capture both global and local geometric details. Methods like FPTransformer, which enhance local feature awareness while maintaining the robustness of transformer architectures, show promise. Techniques that dynamically adjust point-wise features based on local geometric information, like FPConv, also offer significant benefits by preserving local variations and fine details. For sparse point clouds from mesh surface representations of large outdoor scenes, an effective point feature encoding method should be able to handle sparse and uneven point distributions robustly, capture detailed local geometric features without excessive computational overhead, balance global feature aggregation and local detail preservation, and possess high scalability and efficiency for large-scale outdoor data.

3.4 Class Imbalance Problem

The class imbalance problem is a pervasive issue in computer vision, particularly affecting tasks such as object detection, image classification, and segmentation. Class imbalance occurs when certain classes are underrepresented in the training dataset, leading to biased models that perform poorly on minority classes. This problem has spurred numerous research efforts aiming at mitigating its impact:

Loss Function Modification: Modifying loss functions to give more importance to minority classes has been a popular approach. (6) systematically studied different loss functions and their impact on convolutional neural networks (CNNs) under class imbalance conditions and found that using weighted loss functions could help balance the learning process by penalizing the misclassified objectives more heavily or assigning higher weights on rare classes, thus improving the performance on minority classes.

Sampling Techniques: Both oversampling and undersampling methods have been explored extensively. Oversampling minority classes or undersampling majority classes can help balance the dataset. (23) introduced Class Rectification Hard Mining (CRHM), effectively enhancing deep learning performance on imbalanced datasets by selectively sampling difficult instances, which helps in learning more discriminative features.

Gradient Reweighting: Gradient-based methods like gradient reweighting(33) adjust the learning process to compensate for class imbalance. This method has shown promise in class-incremental learning scenarios, addressing both inter-phase and intra-phase imbalances, thereby mitigating the issues of overfitting and catastrophic forgetting.

Data Augmentation: Techniques such as mixup and mosaic augmentations have been shown to improve model performance by creating synthetic samples that balance the dataset. (18) highlighted the effectiveness of these techniques in object detection tasks using the COCO-ZIPF dataset and found that data augmentation methods significantly enhance the mean Average Precision (mAP) by introducing more variability and complexity into the training data.

Cost-Sensitive Learning: Incorporating cost-sensitive measures into the learning algorithm can help mitigate the bias towards majority classes. Fuqua and Razzaghi applies cost-sensitive convolutional neural networks(27) to control chart pattern recognition, which involves adjusting the costs associated with misclassifying minority classes, thus encouraging the model to pay more attention to these classes during training.

Since Focal Loss(47), rebalancing strategies have been at the forefront of addressing class imbalance. Based on the weighting method, these methods could be divided into: 1) Weighting based on the number of category samples, such as EFL(42)/EQL(80)/DECB(108); 2) Weighting based on the gradient ratio of positive and negative samples, such as EQLv2(79); 3) Weighting based on category feature similarity, such as ASCL(88); 4) Weighting based on the number of category samples combined with the number of misclassifications, such as Seesaw Loss(85); 5) Weighting based on mean classification score, such as EBL(25). Based on the parameter setting method, they could be divided into: 1) Static setting parameters, such as EQL(80); 2) Dynamic setting parameters, such as EFL(42)/EQLv2(79)/Seesaw Loss(85)/ASCL(88)/EBL(25)/DECB(108). Based on whether fine-tune is needed, they could be divided into: 1) one-stage, such as EFL(42)/EQL(80)/EQLv2(79)/Seesaw Loss(85)/ASCL(88)/DECB(108); 2) two-stage, such as Distribution Alignment(102)/EBL(25). Composite loss function, such as 1) UFL(99) (dice and cross entropy based)/DFL(35) (FL and DCE);

Each of these techniques comes with its own set of advantages and limitations. Weighting based on the number of category samples offers a straightforward approach by increasing the weight of minority classes to balance the dataset, which is simple and effective but may lead to overfitting in minority classes. Weighting based on the gradient ratio of positive and negative samples dynamically adjusts weights based on training gradients, providing a more nuanced balance but increasing computational complexity. Weighting based on category feature similarity enhances model performance by considering feature similarities between categories, which can improve discrimination but requires sophisticated feature extraction and similarity calculations. Weighting based on both the number of samples and misclassifications offers a balanced approach but adds complexity in tracking misclassifications. Weighting based on mean classification score provides real-time adjustment, improving accuracy, but depends on reliable classification scores. Static parameter setting is simpler to implement and computationally less intensive but lacks flexibility, whereas dynamic parameter setting offers better adaptability and performance but requires complex implementation and longer training times. In terms of fine-tuning, one-stage methods are advantageous for their simplicity and speed, making them suitable for real-time applications, but might not handle severe imbalances as effectively as two-stage methods, which offer refined training stages at the cost of increased complexity. Composite loss functions like UFL and DFL combine multiple strategies, offering robust solutions but accompanied by increased parameter tuning and computational overhead which in some cases can impede the learning of complex loss functions with excessive hyperparameters.. In this paper, we introduce two rebalancing strategies, ACSL and EFL, into our hierarchical loss function, where ACSL dynamically adjusts the weights between categories by considering both feature similarities and the current state of the classifier, enhancing the learning of rare categories without overly suppressing frequent ones. EFL, on the other hand, extends Focal Loss by incorporating category-relevant modulating factors that dynamically balance the contributions of different categories based on their imbalance degrees.

4 Methodology

This chapter delineates the comprehensive methodology proposed, which forms the foundation for subsequent experimental procedures. Section 4.1 presents an overview of the problem to be investigated. Section 4.2 explains the details regarding the feature extraction phase, which incorporates three distinct backbones for the point cloud encoding. The attempts at intricate application of the attention mechanism within the model architecture is thoroughly examined in section 4.3. Finally, section 4.4 alleviates the class imbalance issue prevalent in semantic segmentation tasks and introduces two dynamic reweighting strategies into the hierarchical loss function to mitigate the tendency of current neural networks to ignore rare categories during segmentation.

4.1 Overview

The primary focus of this thesis is to enhance 3D semantic scene reconstruction in large-scale outdoor environments with sparse surface measurements as input by introducing the Semantic Occupancy Network(58) that concurrently accomplishes 3D reconstruction and semantic segmentation by leveraging the advantages of implicit functions. To achieve this, we have optimized the feature encoding structure within the network and alleviated the class imbalance issue inherent in semantic segmentation tasks. As discussed in Chapter 2, the convolutional occupancy network(67) utilizes the occupancy field to implicitly achieve 3D reconstruction. In the semantic occupancy network proposed by(58), the occupied dimension facilitate simultaneous 3D reconstruction and semantic segmentation, thereby resolving boundary inconsistency issues caused by two-step sequential implementation(43)(8).

However, the existing network employs a basic PointNet based variant for point cloud feature extraction and a conventional 3D U-Net for grid feature segmentation. While this design represents a significant advancement from the perspective of pipeline, there remains potential for enhancing the model's performance. This research explores improvements through various feature extraction structures(section 4.2) and incorporates the Swin Transformer(52) into a 3D U-Net like architecture, aiming to making attempts on enhancing semantic segmentation performance using the attention mechanism(section 4.3). Additionally, based on the dynamic gradient adjustments for positive and negative samples, new category dependent reweighting factors are introduced, and based on sample feature similarity, sample gradients are selectively kept, combining with which we try to alleviate the class imbalance problem(section 4.4) in segmentation tasks.

Similar to the structure of the Convolutional Occupancy Network(67), the main difference between the Semantic Occupancy Network(58) and the Convolutional Occupancy Network(67) is that the latter is equivalent to further implementing multi-class segmentation in the occupied space obtained by the former binary classification to achieve simultaneous 3D reconstruction and semantic segmentation. In our pipeline, the input consists of sparse point cloud xyz coordinates extracted from a mesh representation of the scene surface captured by a LiDAR sensor from the air, accompanied by pre-assigned semantic labels. We assume that the majority of the extracted sparse points are located near the scene surface, with only a small portion situated in free space, and subject to some random noise. The output is an occupancy field with predicted class labels, indicating whether each point in the scene belongs to the scene surface and, if so, which of the predefined categories it belongs to.

The following are the basic structural module framework(see figure 4.1) of the Semantic Occupancy Network and an overview of variant combinations of different modules(see table 4.1):

Point Feature Encoding	**Grid Feature Encoding**	**Loss Function**
PointNet based variant	3D U-Net	Original Hierarchical Loss
PointNet++	SwinVFTR	ACSL variant
Deep ConvPN		EFL variant

Table 4.1: Overview of variants' combinations of different modules.

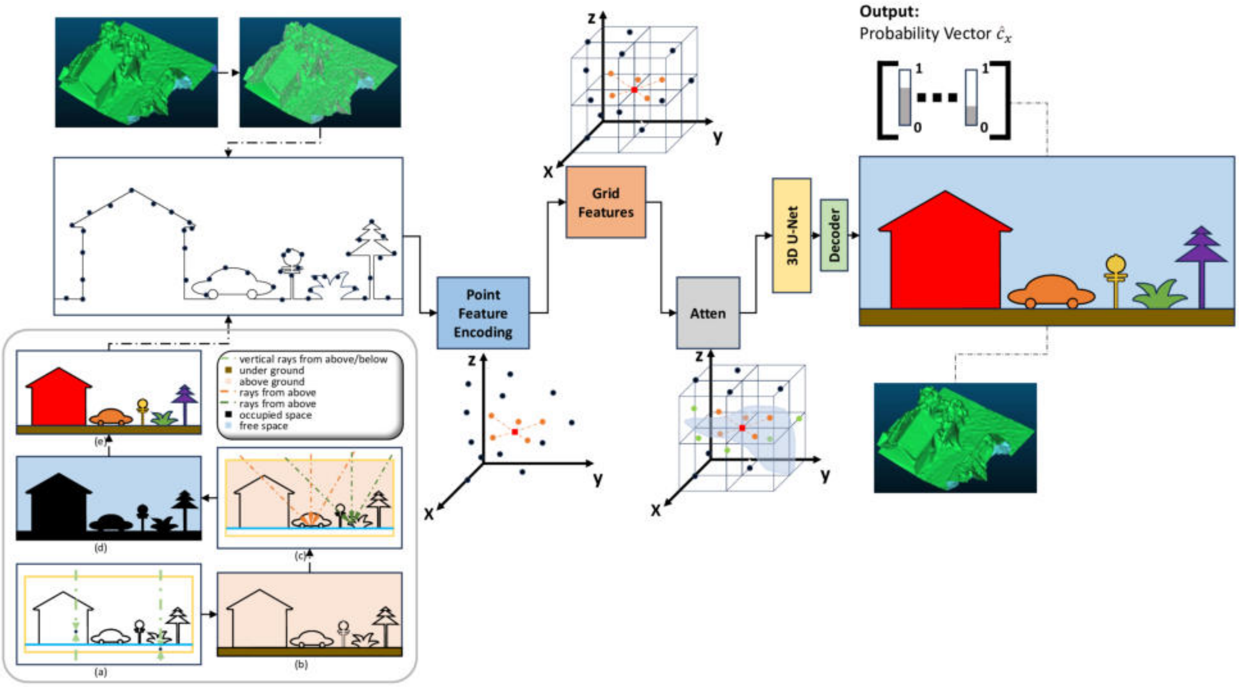


Figure 4.1: Semantic Occupancy Network basic structure. Given the mesh data of a scene, a dense point cloud representing the surface is generated during preprocessing. From this dense point cloud, a sparse subset (1% of the total points in this paper) is selected as the input to the network. Each input point in this subset has xyz coordinates and semantic labels assigned through a two-step preprocessing process. Point feature encoding is applied to these points to generate grid features that fix spatial relationships. An attention mechanism is then selectively employed to focus the model on the features of points near the surface. Then a 3D U-Net like architecture helps in learning voxel-like features. The final stage involves a decoder, composed of simple residual fully connected blocks, which predicts the probability of each category based on the input query points and the encoded features, based on which the final semantic scene surface reconstruction in mesh format is generated. As for the inputs: The input point coordinates are randomly sampled, mostly near the surface, while few of them in free space; The semantic label assignment process is depicted in figures (a)∼(e): (a) uses two rays cast vertically from outside the bounding box to each query point to determine if each point is above or below ground surface, leading to the first classification in (b). For (c), k arbitrary positions above the bounding box (where k is determined by the dataset complexity) serve as origins for rays pointing to each query point. The odevity of the intersections between these rays and the referenced mesh surface indicates whether a point belongs to free space (even number of intersections) or occupied space (odd number of intersections), as shown in (d). Finally, points in the occupied space are assigned specific semantic labels based on the last surface category each ray intersected and the majority of the of the intersected last surface category by k rays, as illustrated in (e). (By the way, the data preprocessing presented here as part of the overview is specific to this paper but not required by the network itself.)

4.2 Point Feature Encoding

4.2.1 PointNet Based Variant

In this work, we explored different point feature encoding strategies for the Semantic Occupancy Network, here the PointNet based variant is the first one. The choice to include a PointNet based variant was motivated by its proven effectiveness in capturing fine-grained details in point cloud data, essential for accurate 3D reconstruction and semantic segmentation tasks.

The PointNet based variant's strengths lie in its robust feature extraction and aggregation capabilities. The local pooling mechanism aggregates local features while preserving geometric details, and multiple ResNet modules enhance the network's ability to learn deep features through residual connections, addressing the vanishing gradient problem. This combination enables the network to capture the intricate details required for precise 3D reconstruction.

However, while the network demonstrates robustness across varying point cloud densities, it remains sensitive to various hyperparameters such as voxel size and hidden dimensions. Compared to the original PointNet structure, which predominantly focuses on global features, this variant slightly increases the emphasis on local features, although there remains potential for improvement.

To further enhance spatial information representation, we propose the integration of advanced positional encoding techniques, including learnable position encodings, attention mechanisms, and dynamic aggregation methods. These enhancements could refine the feature extraction methods and thus improve performance in our 3D reconstruction and semantic segmentation tasks.

In the section of Experiment, an evaluation of the PointNet based variant alongside other point feature encoding methods is involved to establish a comprehensive understanding of their performance in the large-scale outdoor environments. This comparative analysis will help in identifying the optimal feature extraction strategy for the Semantic Occupancy Network, ensuring the results for the semantic scene reconstruction as accurate and reliable as possible.

4.2.2 PointNet++

Motivated by the hierarchical approach for point feature extraction, which captures fine-grained details essential for the dual task of semantic scene reconstruction, PointNet++ was then focused on.

The hierarchical structure of PointNet++ allows it to effectively handle non-uniform point cloud densities. This is achieved through the use of set abstraction layers, each consisting of sampling, grouping, and feature extraction steps as explained in section 2.3.2 . The Multi-Scale Grouping (MSG) and Multi-Resolution Grouping (MRG) techniques further enhance the network's ability to represent features accurately under varying point densities. These capabilities make PointNet++ a strong candidate for our implicit function-based pipeline.

However, the network's computational burden is a significant consideration. Initially, we employ single-scale grouping as the default due to its lower computational requirements. Future work could explore the integration of multi-scale grouping combined with advanced deep convolutional neural networks (e.g., Deep Convpn(41)) to balance computational efficiency and feature extraction performance.

Despite its strengths, PointNet++ is sensitive to the choice of hyperparameters such as the number of sampled points and the radii for neighborhood grouping. Fine-tuning these parameters is crucial to achieve optimal performance. Moreover, while PointNet++ effectively addresses non-uniform sampling density, there is still potential for further enhancement. Integrating learnable position encodings, attention mechanisms, and dynamic aggregation methods could refine the network's ability to capture spatial information, improving performance in 3D reconstruction and semantic segmentation tasks.

4.2.3 Deep ConvPN

To further enrich the feature extraction capabilities of the encoding network from point clouds, and to subsequently integrate attention mechanisms into both the encoding network and a 3D U-Net like structure, we must consider GPU memory consumption alongside performance improvements.

In PointNet++ and its various derivatives, each node duplicates and carries the feature vector information it learns about its neighborhood. This approach leads to significant memory overhead, limiting the network's depth. In the Lean Point Network (LPN)(41), grouping operations are replaced with memory-efficient modules, introducing multi-resolution variants, residual connections, and cross-connections. These enhancements facilitate information flow between layers and across multiple scales, reducing memory consumption while achieving faster training and inference speeds. Consequently, the network can go deeper even with limited memory resources. Additionally, the subsequent integration of attention mechanisms offers an potential to further improve model performance in semantic segmentation tasks.

In ConvPN(41), sequences of pooling modules from single-layer perceptrons (SLPs) replace those of multi-layer perceptrons (MLPs). Within these SLP pooling modules, the multiplications of weight matrices and input points are computed in a single operation. Sparse matrix multiplication is employed to merge zero tensors, thus avoiding explicit duplication of local neighborhood information. This method reduces the storage of numerous intermediate activations, decreasing memory usage. Furthermore, using SLP modules instead of MLPs allows global information to be transmitted earlier in the network compared to PointNet, where global pooling is applied only at the final stage. The sequential composition of SLP modules ensures more frequent sharing of neighborhood information. Utilizing SLP pooling modules and residual connections, a novel convolutional-type point cloud processing module is constructed. Assumed under multi-scale processing scenarios, cross-connections will be introduced to capture information at different resolutions.

The difference between using MLPs and SLPs pooling module(see figure 4.2) is constructed as follows:

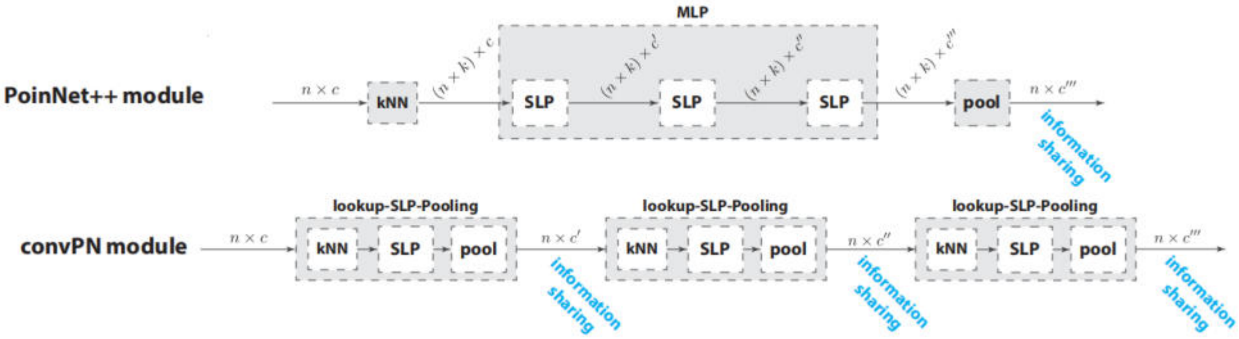


Figure 4.2: A comparison between the architecture of the PointNet++ module and the here proposed convPN module. The convPN module innovatively replaces the Multi-Layer Perceptron (MLP) and its pooling layer with a sequence of Single-Layer Perceptron (SLP)-Pooling modules. This design modification yields two primary benefits: (i) memory savings, as activations are retained only through pooled features, reducing memory footprint, and (ii) improved information flow by enhancing the frequency of information exchange among neighboring points. These advantages contribute to the overall efficiency and effectiveness of the convPN module in point cloud processing.(41)

Since the ConvPN module can be implemented in various network architectures, to make the comparative analysis easier, here we apply it on PointNet++.

After addressing the potential memory limitations, we transitioned from the Single-Scale Grouping (SSG) mode to the Multi-Scale Grouping (MSG) mode and combine it with the Convpn module. The following is the structure of the MSG mode(see figure 4.3) and the current Deep ConvPN network(see figure 4.4):

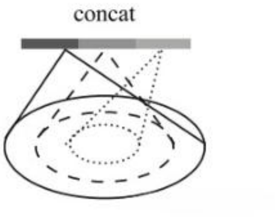


Figure 4.3: Multi-Scale Grouping(MSG)(69)

By adopting the MSG mode, the network can capture multi-scale features more effectively, enabling better performance in complex point cloud processing tasks. The current Deep ConvPN structure incorporates these improvements to ensure efficient feature extraction and propagation, ultimately enhancing the model's performance in semantic segmentation tasks.

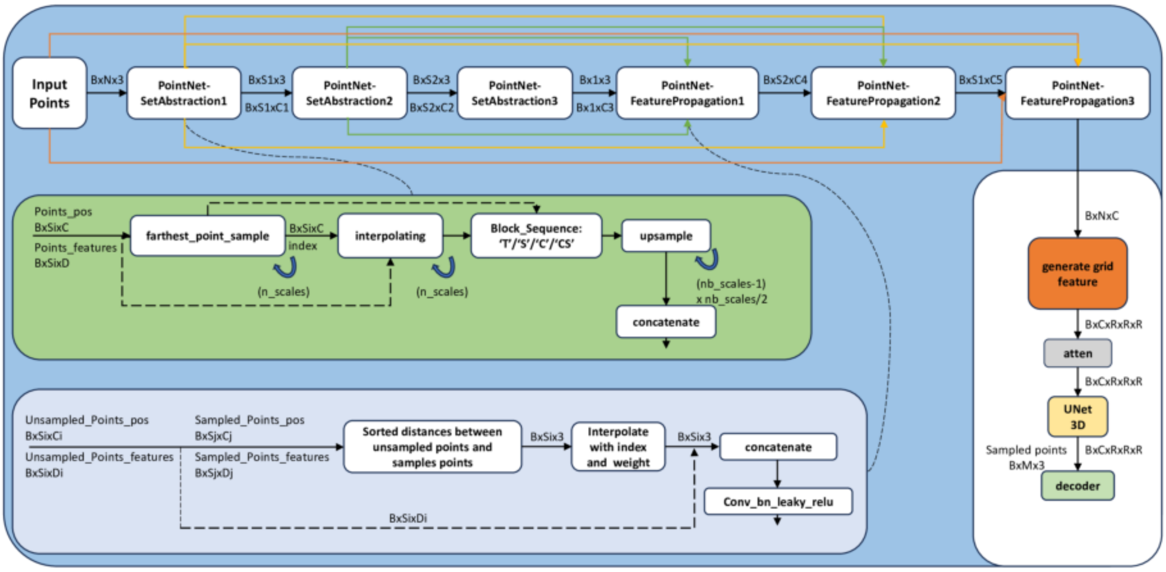


Figure 4.4: Basic structure of Deep Convpn. Here the SLP pooling modules are part of each Block 'T'/'S'/'C'/'CS'.

A preliminary assumption for the feature extraction part could be conducted till now that the integration of memory-efficient modules, multi-scale-grouping variants, and the ability to combine with attention mechanisms will provide a robust framework for improving the performance of point cloud networks while managing GPU memory consumption effectively. These enhancements pave the way for deeper and more efficient networks capable of handling sophisticated 3D reconstruction and semantic segmentation tasks.

4.3 Attention Mechanism in 3D

The attention mechanism facilitates parallel processing of inputs and captures dependencies between any positions within the input sequence. Therefore, we integrate it into the semantic occupancy network(58). Our hypothesis posits that incorporating it in the feature encoding stage can significantly enhance the representational and expressive capabilities of features. This enhancement will improve the model's ability to capture input data details and spatial dependencies, benefiting the accuracy of 3D reconstruction and semantic segmentation tasks. Additionally, combining attention with 3D U-Net allows the attention mechanism to focus on specific regions of interest, enhancing localization capabilities and improving segmentation accuracy. Furthermore, the multi-head attention mechanism can aggregate context information from different parts of 3D volumes, enabling richer context information to guide better segmentation decisions. In sections 4.3.1 and 4.3.2, we introduce novel adaptions for incorporating attention mechanisms into the point feature encoding part and the 3D U-Net (the voxel-like feature encoding part), respectively. These modifications aim to enhance the network's capabilities in the dual tasks of 3D reconstruction and semantic segmentation.

4.3.1 Attention in Point Feature Encoding

Inspired by the architecture of window-based self-attention convolutions(70) and the Swin Transformer(52), we employ self-attention convolutions with varying window sizes to generate features across different frequency bands. Influenced by PET-Neus(89), the use of sliding or moving windows significantly increases the computational complexity of the optimization process. To mitigate this, we attempt to apply single-layer self-attention convolutions and partition the grid features into cells with different window sizes at the location shown in Figure 4.1 after generating the grid features. Due to computational constraints, we are limited to using a window size of 4/2/1 for the point feature encoding stage(where window size 1 means global attention mechanism). The interaction of features at each scale is facilitated by merging features of various scales through subsequent Multi-Layer Perceptrons (MLPs), along with the original features. The concatenated features undergo positional encoding, are dimensionally reduced using MLPs, and are then input into 3D U-Net for further processing.

4.3.2 Attention in 3D U-Net Like Architecture

In this work, we introduce novel methods for incorporating attention mechanisms into the 3D U-Net component of our semantic scene reconstruction pipeline. This includes leveraging the Swin Volumetric Feature-learning Transformer (SwinVFTR) network, which combines SwinTransformer blocks with a 3D U-Net structure as depicted in section 2.4.1 to enhance voxel-like feature extraction and improve performance in semantic scene reconstruction.

The choice to incorporate SwinVFTR was motivated by its hierarchical and multi-scale attention capabilities, which are crucial for capturing fine-grained details and maintaining high-resolution spatial information. By using shifted windows multi-head self-attention (SW-MSA) and Multi-receptive field (MRF) residual blocks, SwinVFTR effectively balances computational complexity with high-quality feature representation. This enables the network to model complex spatial patterns within volumetric data, essential for accurate 3D reconstruction and semantic segmentation.

The SwinVFTR architecture's encoder uses channel-wise volumetric sampling and patch-merging layers to preprocess and downsample input volumes, ensuring compatibility with varying depths and capturing multi-scale information. The decoder employs UnetrUp blocks and volumetric concatenation layers to upsample feature maps and preserve detailed spatial information through efficient skip connections.

A significant innovation of SwinVFTR is the Volumetric Attention (VA) blocks, which apply spatial and channel-wise attention within the skip connections. This enhances the feature maps by emphasizing both spatial and depth information, crucial for accurate semantic scene reconstruction.

4.4 Class Reweighting in Loss Function

Re-weighting classes is an effective method to address the issue of imbalanced classes in the data. By assigning higher weights to the minority classes without adversely affecting the performance on majority classes, the model can focus more on these underrepresented classes during the loss computation. The main advantages of this re-weighting approach are as follows:

Improving Minority Class Recognition: Re-weighting increases the penalty for errors on minority classes during training, compelling the model to better learn the features of these classes.

Balancing the Training Process: Adjusting the weights for different classes can help balance the training process, leading to more uniform performance across all classes and preventing the model from overfitting to the majority classes.

Enhancing Model Robustness: Re-weighting method can improve the model's generalization ability across different classes, making it more stable when dealing with real-world data where class distribution is often uneven.

Overall, re-weighting through the loss function ensures that the model treats all classes more equitably during training, effectively mitigating the negative impact of class imbalance. In section 4.4.1, we firstly introduce the hierarchical loss function defined by the original authors in the semantic occupancy network(58), sections 4.4.2 and 4.4.3 then detail two re-weighting methods for optimizing the latter task of the joint implementation of 3D reconstruction and semantic segmentation in aspect of class imbalance.

4.4.1 Baseline Loss

The semantic occupancy network(58) proposed by Dr. Max Mehltretter incorporates a hierarchical loss function designed to enhance the clarity and effectiveness of spatial categorization. This approach initially classifies space into occupied and unoccupied states. Subsequently, the occupied space is further divided into semantic categories(58). By firstly performing a binary separation, the task of segmentation is simplified, as this binary classification is considered more straightforward than direct segmentation into multiple semantic categories. Integrating binary classification into the semantic segmentation process explicitly improves the loss function, allowing complex tasks to be broken down into simpler components without the necessity for implicit binary segmentation through free space labels.

The hierarchical loss function is formulated as follows:

$$L_H = L_O + \lambda L_S \tag{4.1}$$

where λ balances the contributions of the semantic loss and the occupancy loss.

The semantic segmentation loss is represented by the weighted cross-entropy loss:

$$L_S = -\frac{1}{\sum_{c \in C} w_c} \sum_{x \in X} \sum_{c \in C} w_c y_{x,c} \log(p_{x,c}) \tag{4.2}$$

where w_c is a category-dependent weighting factor, $y_{x,c}$ is a binary indicator for the correct classification of point x in category c, and $p_{x,c}$ is the predicted probability for point x belonging to category c.

The occupancy loss is defined as:

$$L_O = -\sum_{x \in X} \left(o_x \log(\hat{o}_x) + (1 - o_x) \log(1 - \hat{o}_x) \right) \tag{4.3}$$

where o_x is the true occupancy label of point x, and $\hat{o}_x$ is the predicted occupancy probability. o_x equals 0 if the point x is in free space; Otherwise, o_x is 1. The predicted occupancy probability $\hat{o}_x$ is given by $1 - p(freespace)$, with $p(freespace)$ being the predicted probability that x belongs to the free space category.

This structure ensures that the model accurately predicts the presence or absence of objects within a spatial framework and robustly classifies these objects into their respective categories based on learned probabilistic mappings.

4.4.2 EFL Loss

Focal loss(47) is extensively employed to tackle the imbalance between foreground and background in object detection(45)(100) and binary classification tasks(99)(84). It significantly mitigates the influence of the majority background samples by rebalancing the loss contributions from both easy and hard samples.

For binary classification, the focal loss function is given by:

$$\mathrm{FL}(p_\mathrm{t}) = -\alpha_\mathrm{t}(1 - p_\mathrm{t})^\gamma \log(p_\mathrm{t}) \tag{4.4}$$

where p_t denotes the predicted confidence score for the target object, and α_t adjusts the importance between positive and negative samples. The modulating factor $(1 - p_t)^\gamma$ is the core element of focal loss, which diminishes the loss from easily classified samples and emphasizes the learning of hard samples by modulating with the predicted value p_t and the focusing parameter γ. Typically, negative samples are more straightforward to classify, whereas positive samples are more challenging(42). Hence, the imbalance between positive and negative samples can be seen as the imbalance between hard and easy samples(42). The focusing parameter γ influences focal loss: a higher γ value considerably reduces the loss contribution from most negative samples, thereby enhancing the impact of positive samples. Consequently, a greater imbalance between positive and negative samples necessitates a higher γ value.

In multi-class scenarios, focal loss is applied across C classifiers, each handling the output logits transformed by the sigmoid function for every instance. Here, C represents the number of categories, with each classifier treating a specific category as a binary classification task. However, since focal loss uses the same modulating factor for all categories, it fails to address the category imbalance where sample numbers vary significantly.

To alleviate this issue, the Equalized Focal Loss (EFL)(42) function is introduced. EFL incorporates a category-specific focusing factor to address the imbalance between positive and negative samples across various categories.

The EFL loss function for the j-th category is defined as:

$$\mathrm{EFL}(p_{\mathrm{t}}) = -\alpha_{\mathrm{t}}(1-p_{\mathrm{t}})^{\gamma^j}\log(p_{\mathrm{t}}) \tag{4.5}$$

Here, α_t and p_t retain their definitions from the focal loss function, while γ^j is the focusing factor for the j-th category, functioning similarly to γ in focal loss. Different values of γ^j can address the imbalance to varying degrees: a large γ^j can handle severe imbalance in rare categories, while a small γ^j is suitable for frequent categories with slight imbalance. The focusing factor γ^j is split into two components: a category-agnostic parameter γ^b and a category-specific parameter γ_v^j:

$$\gamma^j = \gamma_b + \gamma_v^j = \gamma_b + s(1-g^j) \tag{4.6}$$

where γ^b represents the base focusing factor for balanced data scenarios, while γ_v^j adjusts based on the imbalance degree of the j-th category. Inspired by EQLv2(79), EFL uses a gradient mechanism to select γ_v^j. The parameter g^j indicates the cumulative gradient ratio of positive to negative samples for the j-th category. A large g^j suggests balanced training for the category, while a small g^j indicates imbalance. To fit EFL's requirements, g^j is constrained within [0,1], and $1-g^j$ reverses the distribution. The hyperparameter s determines γ^j's upper limit. Compared to focal loss, EFL independently manages the positive-negative imbalance for each category, improving overall semantic segmentation performance.

Even with class-specific focusing factors, two challenges remain: (1) For binary tasks, a high γ value can alleviate severe positive-negative imbalance. However, in multi-class scenarios, a higher γ reduces the loss contribution, leading to underperformance in rare classes. (2) For small p_t, losses across categories with different focusing factors converge to similar values. Rare hard samples should contribute more loss than frequent hard samples as they can't dominate the entire training process. Therefore, EFL introduces weighting factors to rebalance loss contributions across categories. Similar to the focusing factor, rare categories receive higher weights to boost their loss contributions, while frequent categories' weights remain close to 1. The weighting factor for the j-th category is dynamically set as $\frac{\gamma_b+\gamma_v^j}{\gamma_b}$.The final formula of EFL is formulated as follows:

$$\mathrm{EFL}(p_\mathrm{t}) = -\sum_{j=1}^{C} \alpha_\mathrm{t} \left(\frac{\gamma_b + \gamma_v^j}{\gamma_b} \right) (1 - p_\mathrm{t})^{\gamma_b + \gamma_v^j} \log(p_\mathrm{t}) \tag{4.7}$$

This formulation combines focusing and weighting factors, allowing dynamic loss adjustment based on the category and training status of the sample. When data is evenly distributed, γ_v^j is set to 0, making EFL equivalent to FL. Such application properties enable EFL to be effectively applied to different data distributions and data collectors.

Finally, the semantic loss function variant after incorporating EFL and the complete hierarchical loss function could be expressed as formulas 4.8 and 4.9:

$$\mathcal{L}_{\mathrm{S\_EFL}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} \left(\frac{\gamma_b + \gamma_{v,c}}{\gamma_b} \right) (1 - p_{x,c})^{\gamma_b + \gamma_{v,c}} \log(p_{x,c}) \cdot w_c y_{x,c} \tag{4.8}$$

$$\begin{aligned} \mathcal{L}_{\mathrm{H\_EFL}} = &- \sum_{x \in \mathcal{X}} \left(o_x \log(\hat{o}_x) + (1 - o_x) \log(1 - \hat{o}_x) \right) \\ &+ \lambda \left(\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} \left(\frac{\gamma_b + \gamma_{v,c}}{\gamma_b} \right) (1 - p_{x,c})^{\gamma_b + \gamma_{v,c}} \log(p_{x,c}) \cdot w_c y_{x,c} \right) \end{aligned} \tag{4.9}$$

where o_x means the referenced occupancy state for sample x, indicating whether the sample x is occupied (1) or not (0); $\hat{o}_x$ means the predicted occupancy state for sample x, representing the predicted probability that the sample x is occupied; $\mathcal{X}$ means the whole sample set; $\mathcal{C}$ means the set of all possible classes; λ means the weighting factor for the semantic loss, balancing the importance of the semantic loss with respect to the occupancy loss in the hierarchical loss; $y_{x,c}$ means the referenced label for sample x and class c, which is 1 if sample x belongs to class c, and 0 otherwise; $p_{x,c}$ means the predicted probability that sample x belongs to class c; w_c means the weight for class c, predefined by inversely proportional to the frequency of samples in each class; γ^b represents the base focusing factor for balanced data scenarios; γ_v, c adjusts based on the imbalance degree of the c-th category and could be further decomposed as formula 4.6 by means of dynamic gradient adjustment.

4.4.3 ACSL Loss

In addressing class imbalance, the Adaptive Class Suppression Loss (ACSL)(88) is introduced as another innovative loss function that integrates seamlessly with various datasets without requiring preset groupings or other heuristics. This method offers two significant benefits: Firstly, it prevents rare categories from being excessively suppressed by more frequent categories, and preserves the negative sample gradients for easily confused categories to aid in learning category distinctions. Secondly, ACSL does not rely on label distribution priors, allowing its application across different datasets without recalculating class distribution statistics.

ACSL dynamically selects categories for suppression based on the training process. As depicted in the formula 4.10, a binary weight term w_i is multiplied with the loss term - $\log(\hat{p_i})$ for category i. For a sample belonging to category k, w_i is set to 1. For other categories ($i \neq k$), w_i determines whether suppression is applied based on the output confidence p_i. If p_i exceeds a predefined threshold ξ, indicating confusion between categories i and k, w_i is set to 1 to facilitate discriminative learning. Otherwise, w_i is set to 0 to avoid unnecessary negative suppression. Unlike some previous methods relying on class distribution statistics(80), ACSL leverages the network's output confidence, eliminating the need to find optimal class-statistics-related hyperparameters when switching datasets.

The formula is defined as follows:

$$L_{ACSL}(x_s) = -\sum_{i=1}^{C} w_i \log(\hat{p_i}) \tag{4.10}$$

where

$$w_i = \begin{cases} 1, & \text{if } i = k \\ 1, & \text{if } i \neq k \text{ and } p_i \geq \xi \\ 0, & \text{if } i \neq k \text{ and } p_i < \xi \end{cases} \tag{4.11}$$

The gradient of the loss function with respect to z_i can be expressed as follows:

$$\frac{\partial L_{ACSL}}{\partial z_i} = \begin{cases} p_i - 1, & \text{if } i = k \\ w_i p_i, & \text{if } i \neq k \end{cases} \tag{4.12}$$

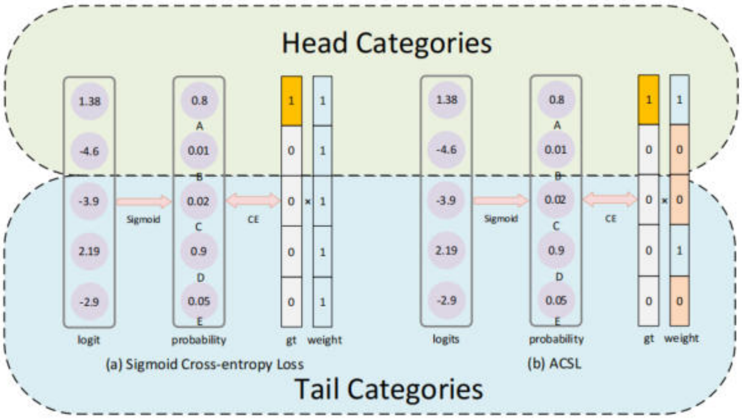


Figure 4.5: ACSL(88). Here is an illustration of Sigmoid Cross-entroy Loss against ACSL, where the top two classes belong to head Categories(frequent and common groups) and the bottom three classes belong to tail categories(rare groups). The predefined threshold ξ should be adapted according to the possibility distribution of categories of each dataset, which in our case is experimentally 0.5.

The accompanying figure 4.5 elucidates ACSL's functionality. Traditional sigmoid cross-entropy loss does not account for imbalanced class distribution, generating negative gradients for all categories (with the weight vector filled with 1), leading to excessive suppression of rare categories and diminished discrimination ability of rare classifiers. ACSL, however, adaptively generates suppression gradients for rare categories. For instance, if the network yields high confidence (e.g., 0.9) for category "D," indicating semantic similarity to category "A," it generates negative suppression for "D," but not for other rare categories with low confidences.

Compared with previous methods, ACSL presents several advantages:

Dynamic Adjustment: ACSL dynamically adjusts the suppression gradient based on the network learning status, unlike methods that pre-adjust based on sample distributions. Similar to EQL(80), it computes a binary weight for each category based on sample counts, which remains constant throughout the training process. ACSL adaptively suppresses categories based on output confidences, resulting in a more efficient learning process.

Fine Granularity: ACSL operates at a finer granularity. Traditional methods apply uniform operations to samples of the same category, such as Class balanced loss(19) and Equalization loss(80), which ignore sample diversity. In contrast, ACSL calculates category weights for each sample based on output confidence, allowing precise control over classification.

Independence from Class Distribution: Many methods rely on label distribution for designing sampling strategies and determining sample weights. This dependency complicates task migration to new datasets. ACSL, however, does not require class frequency priors, facilitating seamless application to new class-imbalanced datasets.

One strategy combined here is to select the number of background samples, here in our case means the unoccupied state(free space), for each category based on their occurrence frequency in the sample set. Compared to the simple approach of dividing categories into frequent, common, and rare groups then assigning a fixed proportion of background samples to each group, the frequency-based method avoids the intermediate classification step and more effectively and intuitively selects the appropriate amount of negative sample suppression for each category, which means proportional to the frequency that each category occurs. The combination of this strategy with Adaptive Class Suppression Loss ACSL) makes it a robust and adaptable solution for managing class imbalance in various deep learning scenarios, including our semantic segmentation task.

$$\mathcal{L}_{\text{S\_ACSL}} = -\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} w_c y_{x,c} \log(p_{x,c}) w_\xi \tag{4.13}$$

$$w_\xi = \begin{cases} 1, & \text{if } c = k \\ 1, & \text{if } c \neq k \text{ and } p_{x,c} \geq \xi \\ 0, & \text{if } c \neq k \text{ and } p_{x,c} < \xi \end{cases} \tag{4.14}$$

$$\begin{aligned} \mathcal{L}_{\text{H\_ACSL}} = & - \sum_{x \in \mathcal{X}} \left(o_x \log(\hat{o}_x) + (1 - o_x) \log(1 - \hat{o}_x) \right) \\ & + \lambda \left(-\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} w_c y_{x,c} \log(p_{x,c}) w_\xi \right) \end{aligned} \tag{4.15}$$

where o_x means the referenced occupancy state for sample x, indicating whether the sample x is occupied (1) or not (0); $\hat{o}_x$ means the predicted occupancy state for sample x, representing the predicted probability that the sample x is occupied; $\mathcal{X}$ means the whole sample set; $\mathcal{C}$ means the set of all possible classes; λ means the weighting factor for the semantic loss, balancing the importance of the semantic loss with respect to the occupancy loss in the hierarchical loss; $y_{x,c}$ means the referenced label for sample x and class c, which is 1 if sample x belongs to class c, and 0 otherwise; $p_{x,c}$ means the predicted probability that sample x belongs to class c; w_c means the weight for class c, predefined by inversely proportional to the frequency of samples in each class; w_ξ means the feature similarity weight for class c, which is 1 if class c is the referenced class, or if $p_{x,c}$ is greater than or equal to the predefined confidence threshold ξ. Otherwise, it is 0; k means the index of the referenced class for a given sample.

5 Experiments

In this chapter, the designs of the experiments are explained in detail. The objectives of the experiments are firstly given by Section 5.1, with a couple of essential questions to be discussed, according to which the whole experimental series are built up. This is followed by a thorough description of the datasets selected for training and testing in Section 5.2, as well as the relevant settings regarding the network in Section 5.3. At last, methods applied for the evaluation of the experimental results are introduced in Section 5.4.

5.1 Objectives

The main task of this thesis is to enhance 3D semantic scene reconstruction in large-scale outdoor environments with sparse surface measurements as input by introducing the Semantic Occupancy Network(58) that concurrently accomplishes 3D reconstruction and semantic segmentation by leveraging the advantages of implicit functions. To achieve this, we have optimized the feature encoding structure within the network and alleviated the class imbalance issue inherent in semantic segmentation tasks as described in detail and comprehensively in Chapter 4. In order to evaluate the effectiveness of this methodology, certain aspects still deserve to be considered. They are summarized into the following questions, according to which the experiments in Chapter 6 are performed and organized:

1) Which feature extraction backbone discussed in this thesis is able to gather more informative features in the encoder part?
2) How effective is the attention mechanism? Where should it be placed within the model structure? Would placing it in the gridding operations of feature extraction or within a 3D U-Net-like architecture, primarily learning the segmentation task, benefit performance enhancement?
3) Could the introduced reweighting method perform better than the original loss function in the aspect of the class imbalance problem?

5.2 Dataset

The dataset builds real measurements from the Hessigheim(39) benchmark. An example illustrating the type of data contained in this dataset is shown in figure 5.1. This benchmark dataset provides 3D triangle mesh-based representations of the German town of Hessigheim for multiple temporal epochs, which have automatically been reconstructed from UAV-based LiDAR measurements and multi-view stereo images. About 60% of the triangles in a mesh have assigned a semantic class label classifying the triangles into 11 classes. As multiple of these classes are surface-related and cannot directly be applied to a volumetric representation, e.g., the facade and the roof of a building cannot clearly be distinguished from a volumetric perspective, we define the following set of semantic classes for our dataset: C_{Hes} = free space, underground, building, vehicle, urban furniture, shrub, tree. The training samples come from 318 cropped samples of the Hessigheim Epoch 2019 March, tested on 50 samples, and evaluated on another 20 samples. The crop size is 32x32x32 m^3. Image rays are employed in the preprocessing step to perform semantic label assignment on noisy point cloud data of this large-scale outdoor scene dataset, as shown in figure 4.1.

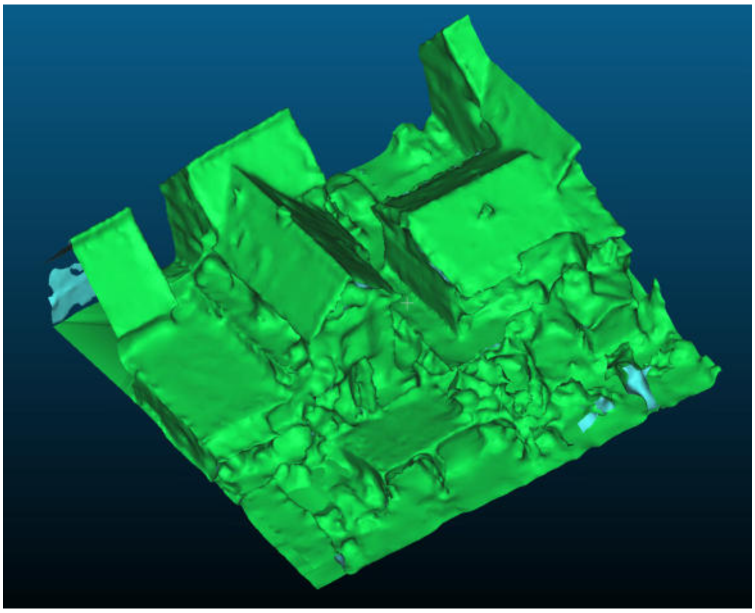


Figure 5.1: An example in mesh format from the Hessigheim dataset.

5.3 Training and Test Settings

The following shows the different parts of the built functional model, including PointNet based variant, PointNet++ and Deep ConvPN for feature encoding, 3D U-Net and SwinVFTR for semantic segmentation tasks, etc. They are all trained on the point cloud version dataset from the mesh format of Hessigheim 3D Epoch 2019 March.

Point Feature Encoding	**Atten**	**Grid Feature Encoding**	**Original**	**ACSL**	**EFL**
PointNet based variant	✓	3D U-Net			
	✗		✓	✓	✓
	-	SwinVFTR			
PointNet++	✓	3D U-Net			
	✗		✓	✓	✓
	-	SwinVFTR			
Deep ConvPN	✓	3D U-Net		✓	
	✗		✓	✓	✓
	-	SwinVFTR		✓	

Table 5.1: Overview of variants' combinations represented in our experiments. Here Atten means a three-layer windowed multi-head self-attention with different window sizes applied at the location after generating grid features as shown in figure 4.1; Original means the original hierarchical loss function, which is the baseline as loss function; ACSL means the baseline embedded with ACSL in semantic part; EFL means the baseline embedded with EFL in semantic part. (*Although all variants were trained, other variants' combinations are not presented in the results for the sake of clarity in the analysis.)

The network structure of all the aforementioned parts is executed on pytoch 2.0.1. Since this is a task of achieving 3D surface reconstruction and semantic segmentation simultaneously, the training process needs to strike a balance between the optimization of the geometric part and the semantic part. However, since this model is combined with the occupancy function, the latter is equivalent to multi-class segmentation based on the former binary classification. During the training process, iou is still used as the learning indicator for optimization, but when tracking the model parameters, iou and mean f1 are tracked at the same time for comparison. It is found that the parameters recorded by the two have a slight difference on the geometric part. Since the iou value itself is high in both cases, in order to alleviate the class imbalance problem, the subsequent experimental results are obtained by testing the model parameters recorded by highest mean f1 values during evaluation. The training was terminated if no improvement was observed in the mean F1 score for 30,000 consecutive iterations. Each batch during the training consisted of sparse point clouds extracted from a 32x32x32 mesh crop representing the scene surface. Specifically, 1% of the points extracted from the mesh were used as inputs to the network. The network parameters were initialized randomly at the beginning of the training.The basic hyperparameters of the model are as follows:

class weights w_c	[1.778, 9.0856, 241.5582, 287.361, 80.7635, 19.993, 3.8856]
crop sizes(m)	32x32x32
percentage input points(%)	1.0
input point cloud noise(m)	0.05
n query points	2048
voxel size(m)	0.5
grid resolution	64
batch size	1
optimizer	Adam
loss weight λ	0.1
learning rate	0.0001

Table 5.2: Basic hyperparameters of the model, where the class weights are used in the formula as depicted in formula 4.9.

5.4 Evaluation Strategies

5.4.1 IoU

The first metric is the Intersection over Union (IOU), which is used to measure the accuracy of predicted occupancy value for the geometric part. IoU provides an intuitive measurement by quantifying the ratio of the intersection over union with regard to the occupancy state in prediction and reference, regarded as an indicator to evaluate the consistency between prediction and reference. Here, we empirically set the prediction value greater than 0.2 as the occupied state. According to the definition of the occupancy field, we set the reference greater than 0.5 as the occupied state. By predicting the intersection over union ratio of the occupied state, we can obtain such a robust index to reflect the reconstruction quality of the geometric part.

The following is the calculation formula for IoU:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{5.1}$$

Here the point sets A and B in the formula 5.1 represent the occupancy states of the query points. Specifically, point set A comprises the reference occupancy values of the query points that exceed 0.5, while point set B consists of the predicted occupancy values that exceed a predefined threshold, which is empirically set to 0.2 in this work.

5.4.2 Overall Accuracy

For the evaluation of semantic segmentation tasks, we firstly introduce overall accuracy, which is a core evaluation metric used to quantify the overall performance of the model in predicting the semantic label of each query point that randomly sampled from the input sparse point cloud. This indicator evaluates the overall effect of the model by calculating the consistency between the predicted labels and the reference labels. Here, the valid predictions and reference points are firstly filtered out through the semantic reference valid mask to avoid the potential issue of points that may not have been correctly assigned semantic labels during the preprocessing of the point cloud, and then the ratio of the correctly predicted points to the total number of valid sample points is calculated to obtain the overall accuracy.

Here is the formula for calculating overall accuracy:

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FP_i + FN_i + TN_i)} \tag{5.2}$$

5.4.3 Mean F1 Score and Per Class F1

Mean F1 Score and Per Class F1 Score are another two important indicators to measure model performance in the semantic segmentation task. These two metrics evaluate the model's accuracy and consistency across categories by calculating the harmonic mean between predicted point labels and referenced point labels. Among them, Per Class F1 Score is used to calculate the F1 Score of each category. This indicator combines precision and recall, providing a detailed analysis of the model's performance on a single category. For classification tasks with long-tail distribution of complex datasets and semantic segmentation tasks of large outdoor scenes discussed in this paper, there widely occurs the class imbalance problem, and the analysis by means of these two indicators is of great significance. The Mean F1 Score is the average of the F1 scores for all categories. This average provides a global perspective on the overall performance of the model on all categories. These two indicators are very beneficial to the performance optimization evaluation of semantic segmentation tasks.

The following are the calculation formulas of F1 Score:

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$

where

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.5}$$

$$\text{Mean F1 Score} = \frac{1}{n} \sum_{i=1}^{n} \text{F1 Score}_i \tag{5.6}$$

5.4.4 Chamfer Distance

In order to quantify the difference between the geometric position of the mesh generated by the occupancy network prediction and the geometric position of the reference points, we introduce chamfer distance as an evaluation metric to evaluate the accuracy of the generated mesh. Generally speaking, chamfer distance is a measurement of the difference between two point sets, which calculates the average distance from the predicted point set to each nearest point in the reference point set, and the average distance from the reference point set to each nearest point in the predicted point set. Here, after visualized mesh format data is generated from the predicted points in the test, we turn off the visualization generation, and then use chamfer distance to evaluate the accuracy of the generated mesh. As an evaluation indicator, chamfer distance has many advantages. It is a bidirectional calculation of the distance between two point sets. It can effectively measure the overall geometric similarity of the shape without being affected by the scale of the point set. It is very conducive to evaluating the degree of agreement between 3D mesh generated from the predicted points of the occupancy network and the actual referenced points. In addition, it is robust to point clouds with different densities and sampling methods.

Here is the formula for calculating chamfer distance:

$$\text{Chamfer Distance} = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2 \tag{5.7}$$

6 Results

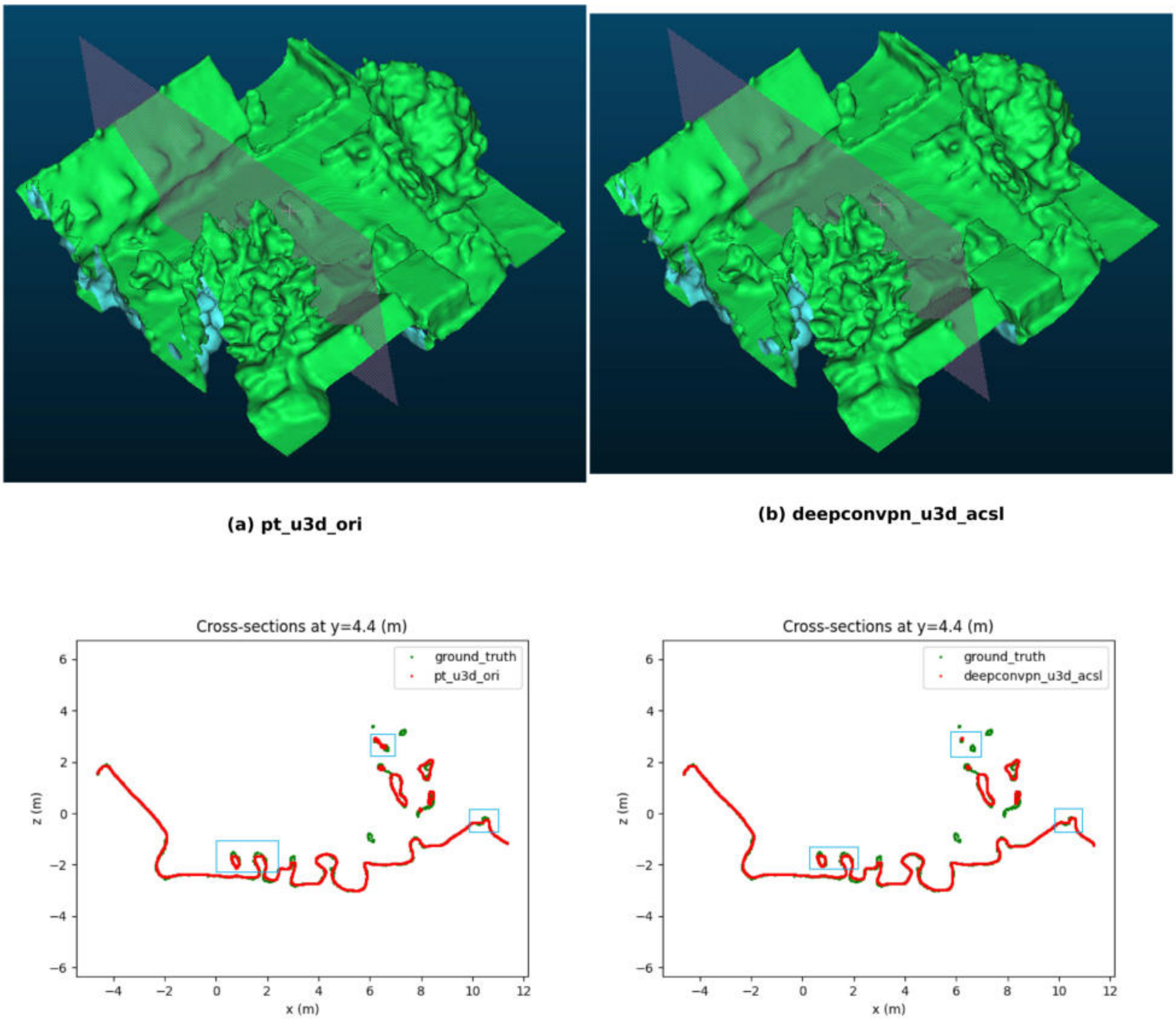
This chapter analyzes the experimental results, systematically addresses and substantiates the questions raised in Chapter 5 through a detailed data-driven approach. The analysis is divided into four sections, each providing evidence that discusses the validity of the proposed method, thereby demonstrating its overall effectiveness.

Comparison between The Baseline and Our Proposed Framework

As shown in Table 6.1, we could observe differences across various metrics — iou, oa, mean f1, and chamfer distance — when comparing different variants. Relative to the baseline, our proposed framework achieves 5.00% improvement in the mean f1 score, which menas better performance on the semantic subtask. Although our framework was not specifically optimized to track the best iou score as the primary performance metric during training, it still demonstrates a slight improvement in iou. While the Deep ConvPN/u3d/efl variant marginally outperforms our proposed framework in terms of iou, we argue that the current Deep ConvPN/u3d/acsl framework performs better due to potential limitations of the former, which will be discussed in the third part.

	iou(%)	**oa(%)**	**mean f1(%)**	**free space(%)**	**building(%)**	**shrub(%)**	**urban furniture(%)**	**vehicle(%)**	**tree(%)**	**underground(%)**	**chamfer distance(m)**
pointnet/u3d/original(baseline)	89.15	93.41	58.55	96.46	72.66	28.44	11.50	16.21	52.29	92.62	0.086
pointnet/u3d/acsl	90.75	93.79	63.37	96.82	75.08	46.88	16.35	15.78	54.65	92.23	0.086
pointnet/u3d/efl	90.11	93.68	61.65	96.95	74.47	43.79	19.60	14.77	51.71	91.74	0.071
Deep ConvPN/u3d/original	90.00	93.64	61.09	96.65	68.24	37.04	15.66	18.17	53.10	92.26	0.059
Deep ConvPN/u3d/acsl(proposed)	90.41	93.97	61.48	96.85	74.87	42.00	16.37	16.44	55.20	92.56	0.088
Deep ConvPN/u3d/efl with best s	90.15	93.78	61.85	96.91	66.43	32.12	18.26	18.60	58.85	91.89	0.088

Table 6.1: Comparison of different variants on various evaluation metrics. Here the values in blue means they are higher than the baseline, while the values in red mean they are the highest in each column.



(a) pt_u3d_ori

(b) deepconvpn_u3d_acsl

(c) cross_section_of_pt_u3d_ori

(d) cross_section_of_deepconvpn_u3d_acsl

Figure 6.1: The comparison between the baseline ((a) pt_u3d_ori) and our proposed framework ((b) deepconvpn_u3d_acsl). Figures (a) and (b) display the surfaces generated by the marching cubes algorithm from the resulted implicit fields produced by the respective methods. Figures (c) and (d) show the cross-section of the same position along the y-axis for the baseline and our proposed framework respectively(red), while the green line represents the cross-section from the ground truth for the corresponding scene. The differences between (a) and (b) are highlighted by the blue boxes.

As illustrated in Figure 6.1, our proposed framework (deepconvpn_u3d_acsl) demonstrates a clear advantage over the baseline in reconstructing fine details, such as the geometric features of shrubs and rooftop chimneys, as well as some smaller geometric structures. This superiority can be attributed not only to the more frequent weight sharing, faster feature propagation, and enhanced focus on local details within the deepconvpn encoding module, but also to the acsl variant's ability to adjust sample frequencies for rare categories and filter out similar feature classes. However, Figure 6.1 also reveals that for certain small geometric objects (e.g., portions of categories within the cross-section where the ground truth, represented by the green line, is fully exposed), the deepconvpn_u3d_acsl framework almost completely overlooks them, which indicates that further improvements are needed to better handle such irregular geometries.

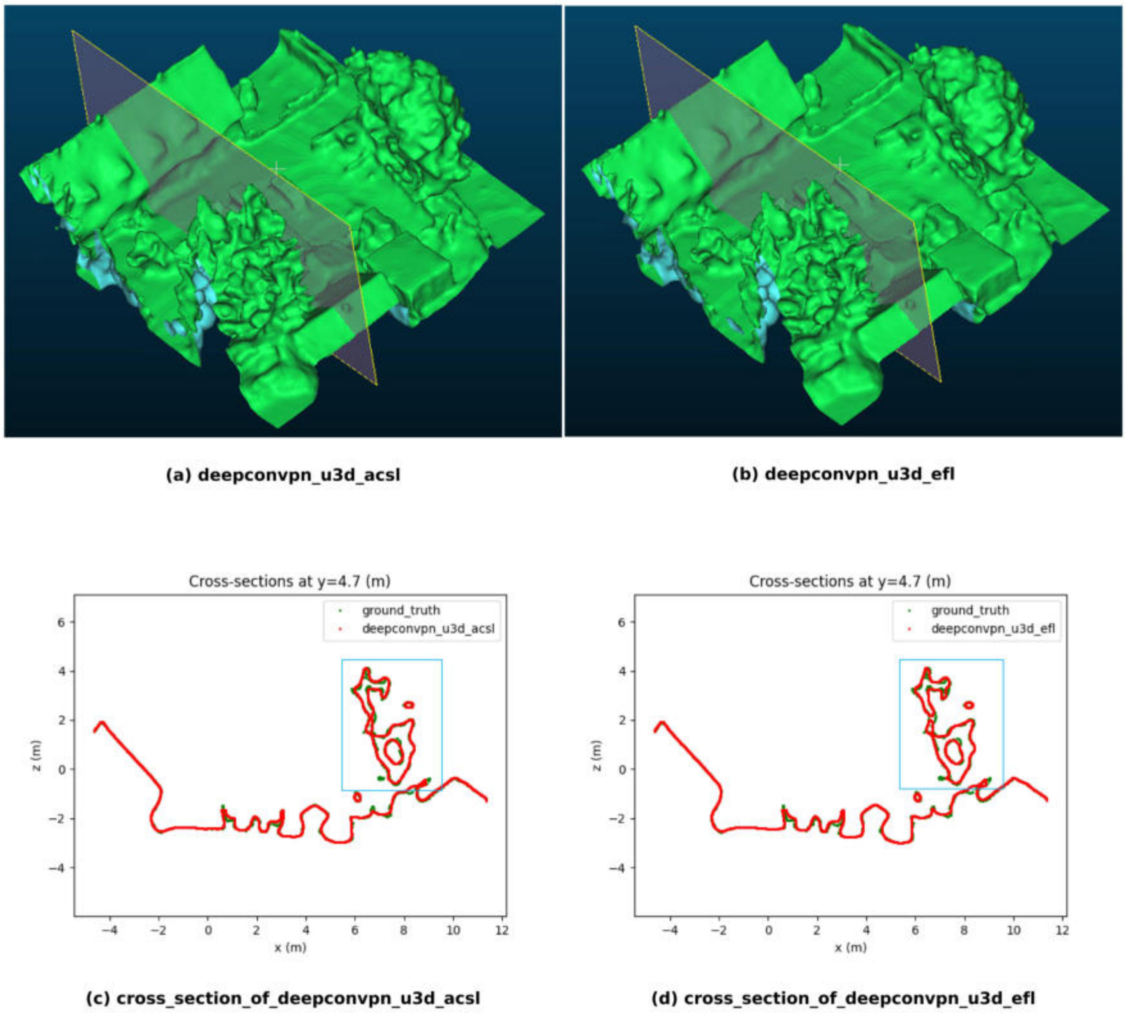
Comparison among Different Point Feature Encoding Methods

	iou(%)	**oa(%)**	**mean f1(%)**	**free space(%)**	**building(%)**	**shrub(%)**	**urban furniture(%)**	**vehicle(%)**	**tree(%)**	**underground(%)**	**chamfer distance(m)**
Deep ConvPN/u3d/original	90.00	93.64	61.09	96.65	68.24	37.04	15.66	18.17	53.10	92.26	0.059
Pointnet++/u3d/original	89.26	93.34	60.68	96.43	72.37	45.01	14.96	15.62	52.46	92.23	0.059
Pointnet/u3d/original(baseline)	89.15	93.41	58.55	96.46	72.66	28.44	11.50	16.21	52.29	92.62	0.086

Table 6.2: Comparison of model structure with different point feature encoding. Here the values in blue means they are higher than the baseline, while the values in red mean they are the highest in each column.

Table 6.2 compares the performance metrics of Deep ConvPN, PointNet++, and PointNet based variant as point feature encoding structures, while keeping other components unchanged. From this comparison, it is evident that Deep ConvPN demonstrates superior performance in terms of iou, overall accuracy(oa), and mean f1 score. Additionally, as shown in Table 6.2, Deep ConvPN improves segmentation accuracy for minority classes 2, 3, 4, and 5. Furthermore, the chamfer distance analysis indicates that even under conditions prioritizing mean f1 score during validation, Deep ConvPN as a backbone delivers equally high-quality mesh reconstruction results compared to PointNet++.

Comparison between The Two Reweighting Loss Function Strategies



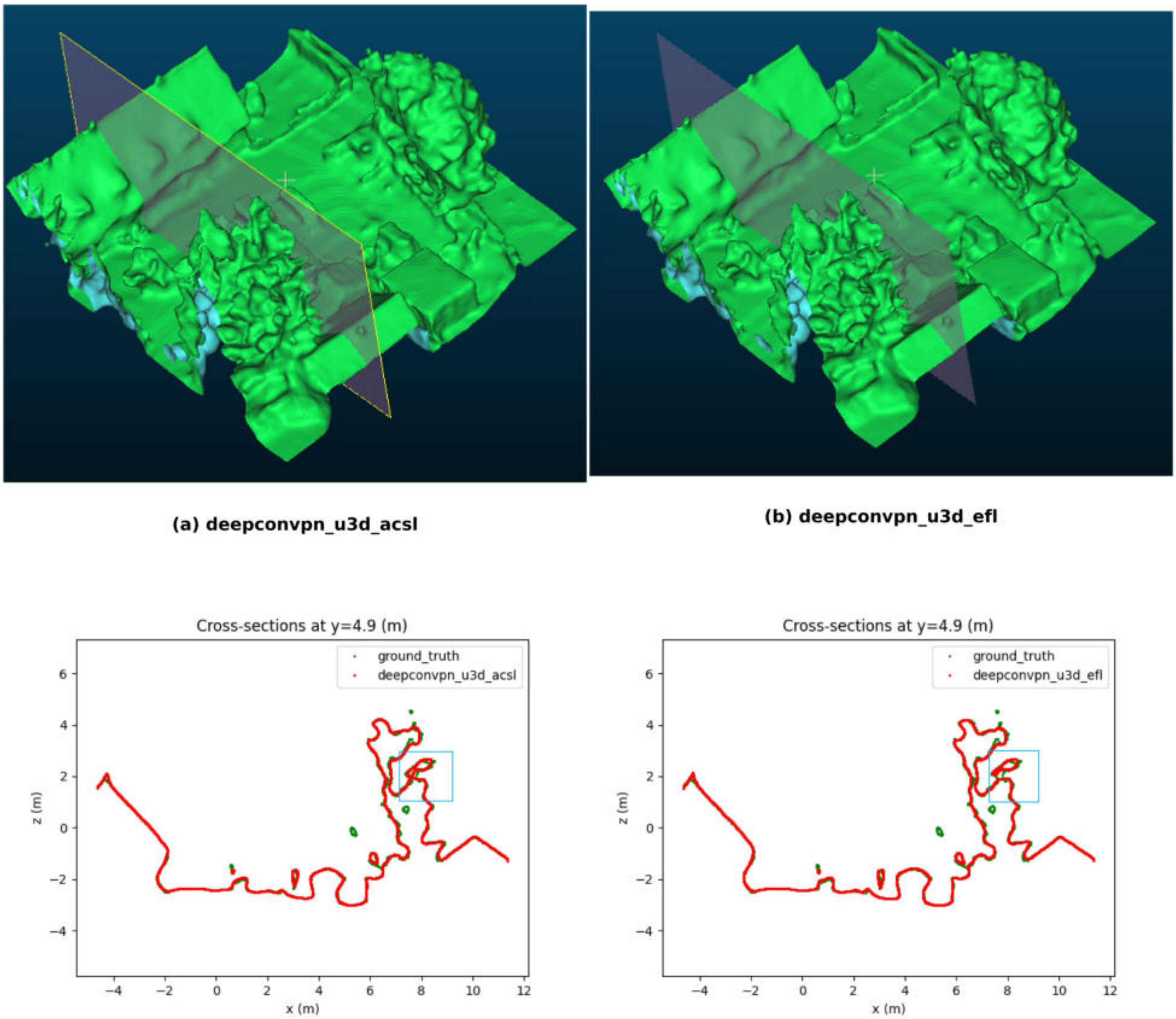
(a) deepconvpn_u3d_acsl

(b) deepconvpn_u3d_efl

(c) cross_section_of_deepconvpn_u3d_acsl

(d) cross_section_of_deepconvpn_u3d_efl

Figure 6.2: The comparison between the acsl variant ((a) deepconvpn_u3d_acsl) and the efl variant ((b) deepconvpn_u3d_efl). Figures (a) and (b) display the surfaces generated by the marching cubes algorithm from the resulted implicit fields produced by the respective methods. Figures (c) and (d) show the cross-section of the same position along the y-axis for (a) and (b)(red), while the green line represents the cross-section from the ground truth for the corresponding scene. The differences between (a) and (b) are highlighted by the blue boxes.



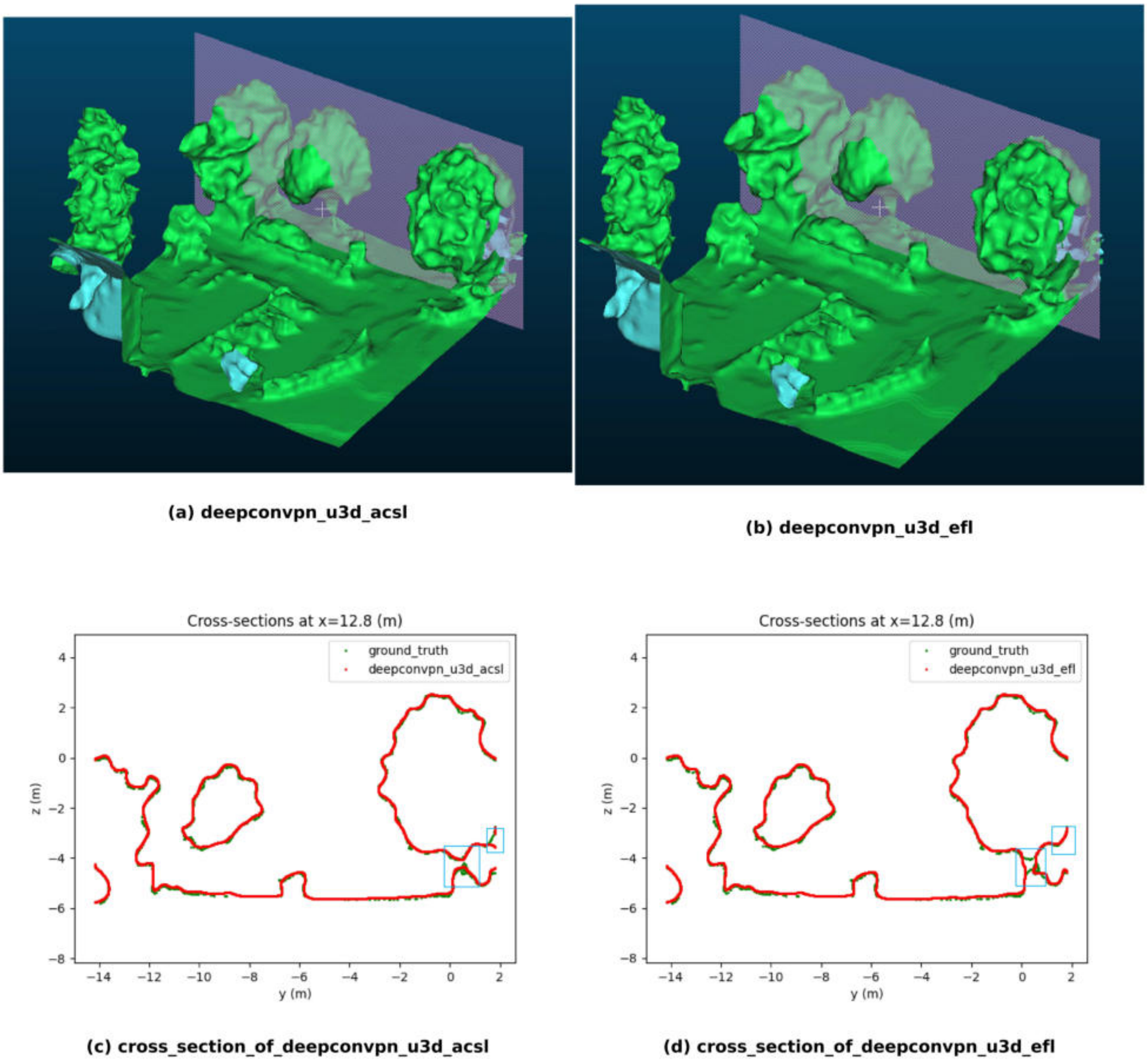
(a) deepconvpn_u3d_acsl

(b) deepconvpn_u3d_efl

(c) cross_section_of_deepconvpn_u3d_acsl

(d) cross_section_of_deepconvpn_u3d_efl

Figure 6.3: The comparison between the acsl variant ((a) deepconvpn_u3d_acsl) and the efl variant ((b) deepconvpn_u3d_efl). Figures (a) and (b) display the surfaces generated by the marching cubes algorithm from the resulted implicit fields produced by the respective methods. Figures (c) and (d) show the cross-section of the same position along the y-axis for (a) and (b)(red), while the green line represents the cross-section from the ground truth for the corresponding scene. The differences between (a) and (b) are highlighted by the blue boxes. Here, we present the same example as in Figure 6.2 but a different position along the y axis.



(a) deepconvpn_u3d_acsl

(b) deepconvpn_u3d_efl

(c) cross_section_of_deepconvpn_u3d_acsl

(d) cross_section_of_deepconvpn_u3d_efl

Figure 6.4: The comparison between the acsl variant ((a) deepconvpn_u3d_acsl) and the efl variant ((b) deepconvpn_u3d_efl). Figures (a) and (b) display the surfaces generated by the marching cubes algorithm from the resulted implicit fields produced by the respective methods. Figures (c) and (d) show the cross-section of the same position along the x-axis for (a) and (b)(red), while the green line represents the cross-section from the ground truth for the corresponding scene. The differences between (a) and (b) are highlighted by the blue boxes. Here is another example different from Figure 6.2.

Figures 6.2, 6.3, and 6.4 illustrate examples from two cropped scenes. From these figures, we could observe that the variants of both rebalancing strategies achieve good reconstruction results for classes of relatively regular primitives, such as buildings and vehicles, with cross-sections closely matching the ground truth, indicated by the green line. However, for irregular geometric objects like shrubs and trees, both variants tend to make errors in detailed reconstruction. This issue is particularly pronounced in the efl variant, where single geometric structures are often reconstructed as two separate entities, as seen in Figures 6.2, 6.3, and 6.4. Although the acsl variant performs significantly better in this regard, it still exhibits a similar issue in Figure 6.4, where the boundary of a tree is reconstructed as discontinuous. This problem arises partly due to the imbalanced class distribution in the dataset and the sparse input, which results in insufficient samples for effective learning during training. Additionally, the irregularity of trees and shrubs exacerbates the challenge of sparse training, suggesting that it may be necessary to impose additional geometric constraints on irregular classes to enhance their representation in large-scale sparse scenes. Moreover, setting aside the potential issues related to the dataset and irregular geometries, the relatively better performance of the acsl variant may stem from its adaptive background sample selection. Unlike the efl variant, which relies on preset scale parameters that might overly emphasize certain classes, the acsl variant adapts the focus on classes by adjusting to both background and foreground samples. This prevents rare classes from being overwhelmed by the negative gradients of background samples and reduces the similarity features that need to be learned across classes, potentially making it easier for the model to learn features for classes with limited samples.

	iou(%)	**oa(%)**	**mean f1(%)**	**free space(%)**	**building(%)**	**shrub(%)**	**urban furniture(%)**	**vehicle(%)**	**tree(%)**	**underground(%)**	**chamfer distance(m)**
1	90.09	93.83	60.58	96.90	73.26	36.61	19.27	13.89	51.43	92.03	0.059
2	90.28	93.54	59.62	96.98	69.09	36.55	18.75	15.78	48.82	91.47	0.087
3	89.61	93.59	57.43	96.97	68.79	33.22	14.85	13.51	48.61	92.10	0.080
4	90.11	93.68	61.65	96.95	74.47	43.79	19.60	14.77	51.71	91.74	0.071
6	88.95	93.62	59.38	96.79	71.84	42.97	16.34	15.34	50.23	92.46	0.087
8	89.08	93.66	60.76	96.96	71.73	47.61	16.90	19.58	50.34	91.59	0.058

Table 6.3: Adjustment of s(scale factor) for baseline embedding EFL into the original hierarchical loss function. Here blue means higher than baseline, red indicates perform best in each column.

Table 6.3 and 6.4 illustrate the impact of different scale factors on various performance metrics of the model after integrating Equalized Focal Loss(EFL) into the original hierarchical loss function as formula 4.9. Since EFL primarily mitigates the model's issue ignoring minority classes, the baseline(pt/u3d/ori) embeded with EFL achieves optimal performance with a scale factor when s=4. In contrast, for the combination of Deep ConvPN/u3d/efl, the best balanced segmentation accuracy across all categories is achieved with a scale factor when s=6. Although the potential issues of overfitting rare categories and insufficient representation of frequent categories, which can arise from increasing the scale factor to enhance focus on rare categories, were not evident in this study, adjusting the scale factor still requires a significant amount of additional time.

	iou(%)	oa(%)	mean f1(%)	free space(%)	building(%)	shrub(%)	urban furniture(%)	vehicle(%)	tree(%)	underground(%)	chamfer distance(m)
1	90.22	93.65	61.04	96.87	71.89	35.90	17.03	14.88	54.91	91.51	0.092
2	90.11	93.91	58.99	96.88	70.89	49.53	15.81	16.97	52.04	92.38	0.086
3	89.97	93.86	59.63	96.95	70.85	31.89	18.35	16.80	51.05	91.68	0.088
4	89.69	93.62	58.01	96.87	68.98	40.44	13.25	17.46	50.88	91.99	0.088
6	90.15	93.78	61.85	96.91	66.43	32.12	18.26	18.60	58.85	91.89	0.088
8	89.44	94.04	60.92	96.92	65.06	37.54	19.67	19.27	55.56	92.20	0.090

Table 6.4: Adjustment of s(scale factor) for model combination of Deep ConvPN/u3d/efl embedding EFL into the original hierarchical loss function. Here blue means higher than baseline, red indicates perform best in each column.

Class	**Frequency [%]**
class 0 - free space	56.30%
class 1 - building	11.00%
class 2 - shrub	1.20%
class 3 - urban furniture	0.30%
class 4 - vehicle	0.40%
class 5 - tree	5.00%
class 6 - underground	25.70%

Table 6.5: Occurrence of each category in the training set.

Evaluation Of The Attention-related Application

Leveraging the advantages of attention mechanisms, we attempted to integrate single or multiple multi-head window self-attention mechanisms of varying sizes into the grid features generated after point feature encoding, specifically before the voxel-like feature encoding in the 3D U-Net-like structure. However, as evidenced in Table 6.6, the inclusion of such attention mechanism, for instance, a three-layer multi-head window self-attention mechanism at the location as shown in figure 4.1, did not result in any improvement in either geometric or semantic accuracy. One potential reason could be that the integration of semantic information caused an ineffective fusion of numerous pieces of information between the two subtasks, leading to unknown interferences. This is corroborated by prior experiments with pure geometric tasks using convolutional occupancy networks, where different attention mechanisms were experimentally added at the same position before modifying the loss function. Although the PointNet based variant, serving as the baseline, has limited feature encoding capabilities, a slight performance improvement in the preliminary attempts is observed on purely geometric tasks after embedding the attention mechanism.

Another possible cause could be the insufficient hyperparameters' adaption based on the hardware resources, such as the limited window size and batch size. All experiments in this study were conducted on an RTX 4090 GPU, and memory collapse occurred when the window size was too large. The insufficient capability of the feature processing methods might also contribute to this issue, but switching the backbone from a PointNet based variant to Deep ConvPN did not resolve this limitation, indicating that other feature processing methods might be required.

We also considered the potential inapplicability of the window self-attention mechanism and the U-Net architecture. Thus, we introduced SwinVFTR, a model architecture that combines Swin Transformer and 3D U-Net, recently applied in brain tumor segmentation tasks. However, as shown in Table 6.6, despite a slight improvement in the geometric component (IoU), there was a significant loss in segmentation accuracy. The possible reason for this could be that although Swin-VFR combines the sliding window module of Swin Transformer with the symmetric structure of 3D U-Net, enabling effective aggregation of both local and global contextual features, it simultaneously increases the model's parameters and complexity manifold, potentially leading to instability and insufficient optimization during training. Additionally, the large outdoor dataset used in this thesis may have had insufficient data within minority classes, preventing the model from learning effective features.

	iou(%)	oa(%)	mean f1(%)	free space(%)	building(%)	shrub(%)	urban furniture(%)	vehicle(%)	tree(%)	underground(%)	chamfer distance(m)
Deep ConvPN+atten/u3d/acsl	90.20	93.42	58.84	96.81	72.56	31.95	15.31	14.27	49.24	92.24	0.090
Deep ConvPN/swinvftr/acsl	90.68	92.93	57.48	96.75	61.20	35.28	14.31	15.09	47.57	91.99	0.090
Deep ConvPN/u3d/acsl	90.41	93.97	61.48	96.85	74.87	42.00	0.1637	16.44	55.20	92.56	0.088

Table 6.6: Comparison of variants embedded with different attention mechanism. Here blue means higher than baseline, red indicates perform best in each column.

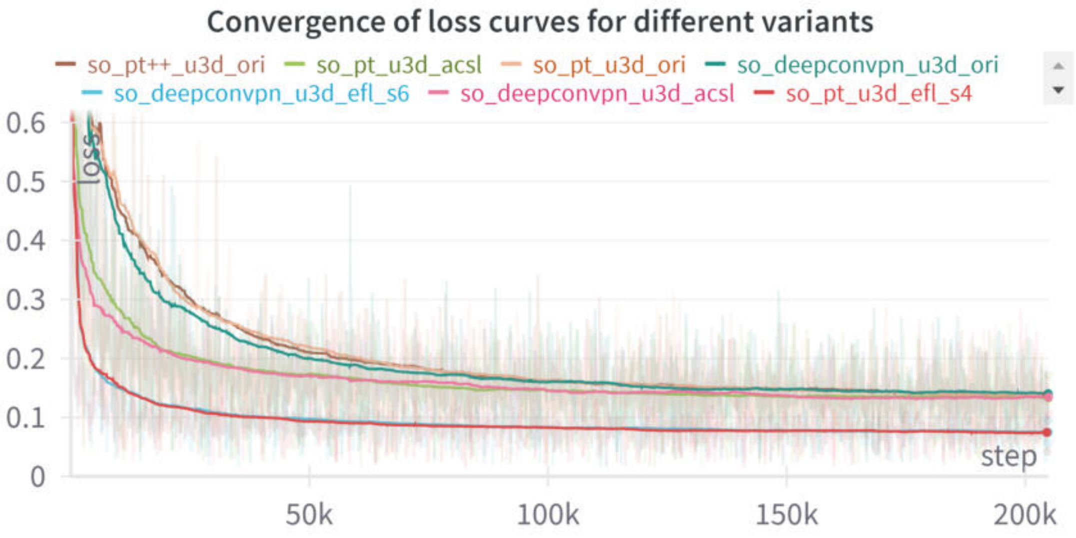


Figure 6.5: Convergence of loss curves for different variants.

The loss curves depicted in Figure 6.5 illustrate that the model variants utilizing Equalized Focal Loss (EFL) and Adaptive Class Suppression Loss (ACSL) exhibit superior performance in terms of loss reduction and stability. These losses are particularly advantageous for long-tailed distributions in real world scenarios and segmentation tasks, enabling more effective optimization of model parameters and enhancing overall training efficacy.

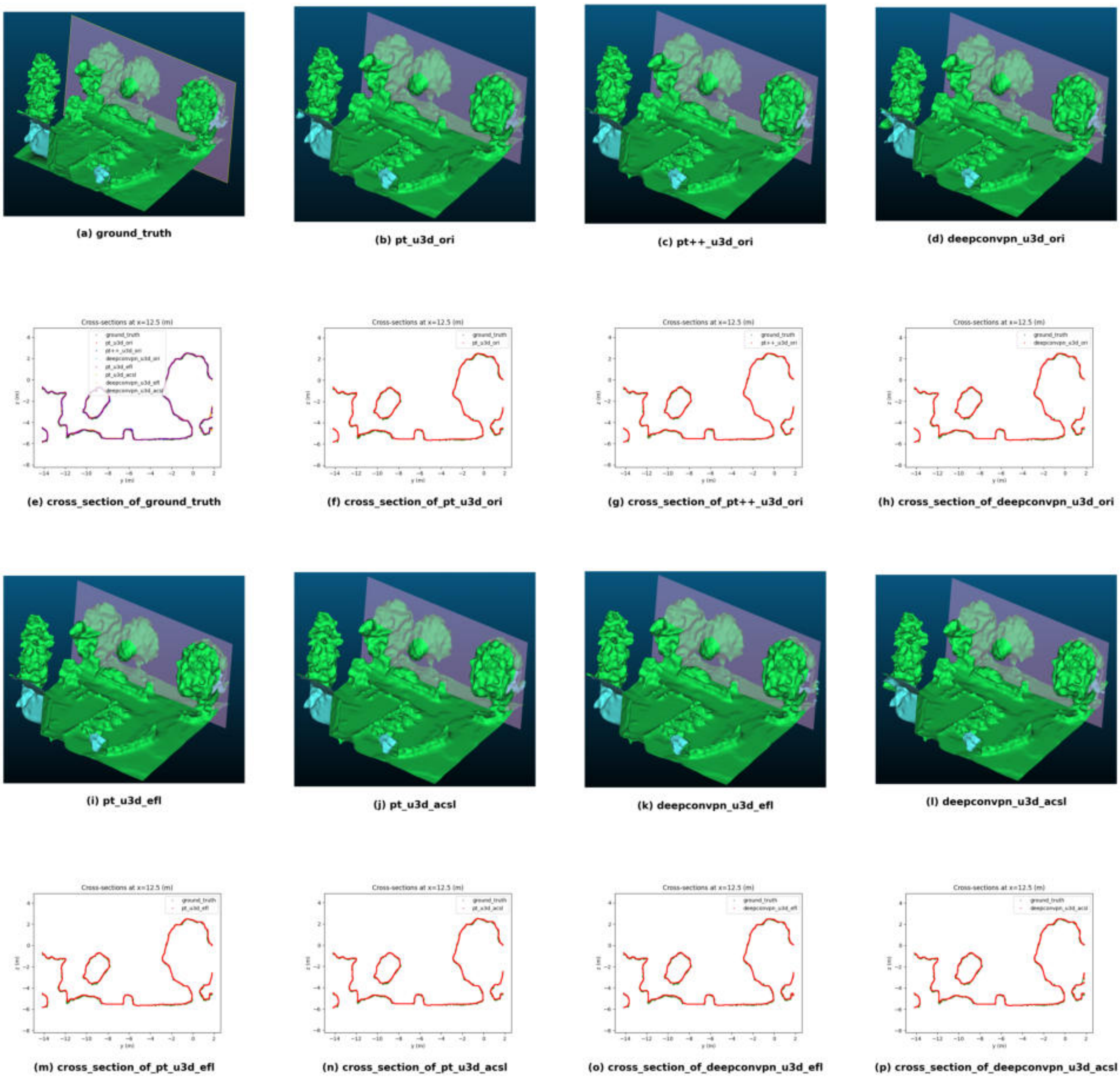
7 Conclusion and Outlook

This paper integrates the PointNet based variant and subsequent researched modules built upon it into the Semantic Occupancy Network to optimize point feature encoding structures, aiming to enhance both geometric and (mainly) semantic performance in this dual-task framework. Additionally, novel reweighting loss functions are embedded to address class imbalance issues inherent in the simultaneous implicit semantic scene surface reconstruction, thereby improving segmentation accuracy for minority categories.

Experimental results demonstrate that integrating Deep ConvPN as the point feature encoding module with 3D U-Net and rebalancing strategies could significantly improve both geometric and semantic accuracy in the field of semantic scene surface reconstruction in large-scale outdoor environments with sparse point cloud as inputs. Specifically, Deep ConvPN enhances the baseline by replacing the MLP module with the SLP pooling modules, which introduces early weight sharing and increases the frequency of this sharing. However, the expected performance improvements from embedding the Attention mechanism in both the point feature encoding module and the voxel-like feature encoding module are not observed. This lack of improvement may stem from unforeseen interferences that arise when semantic information is combined with geometric tasks, and the specific impacts of this require further investigation. Meanwhile, the Adaptive Class Suppression Loss(ACSL) and Equalized Focal Loss(EFL), which can be seamlessly integrated into various networks to mitigate class imbalance, do indeed deliver the anticipated improvements in segmentation accuracy for minority classes when incorporated into our hierarchical loss function. However, EFL necessitates additional tuning for the scale factor, which introduces potential time costs.

Future research could design further geometric constraints like using implicit algebraic computation for irregular geometries such as bushes and trees to aid in high-precision 3D surface reconstruction. While introducing additional technologies to improve the accuracy of simultaneous tasks, it is crucial to consider the added complexity, which may impede real-time responsiveness. Finally, the potential application of other attention mechanisms in this pipeline should be explored under the condition of unconstrained computational resources to adapt the hyperparameters. Additionally, it is also possible to embed a KAN module, which leverages external knowledge bases to enhance the model's contextual awareness, thereby improving its ability to handle visual tasks.

Appendix



(a) ground_truth

(b) pt_u3d_ori

(c) pt++_u3d_ori

(d) deepconvpn_u3d_ori

(e) cross_section_of_ground_truth

(f) cross_section_of_pt_u3d_ori

(g) cross_section_of_pt++_u3d_ori

(h) cross_section_of_deepconvpn_u3d_ori

(i) pt_u3d_efl

(j) pt_u3d_acsl

(k) deepconvpn_u3d_efl

(l) deepconvpn_u3d_acsl

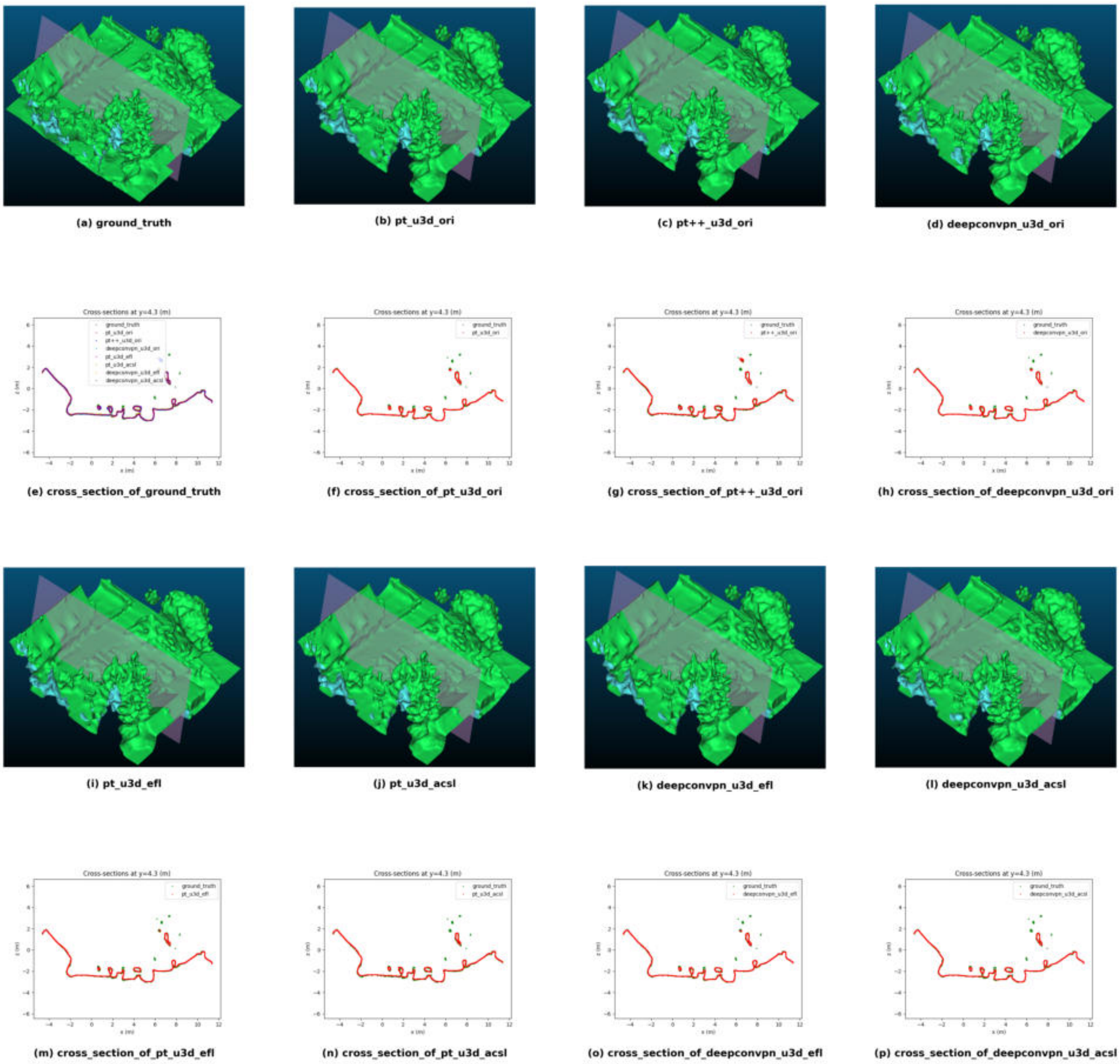
(m) cross_section_of_pt_u3d_efl

(n) cross_section_of_pt_u3d_acsl

(o) cross_section_of_deepconvpn_u3d_efl

(p) cross_section_of_deepconvpn_u3d_acsl

Figure 1: Representations of different variants on category tree and vehicle.



(a) ground_truth

(b) pt_u3d_ori

(c) pt++_u3d_ori

(d) deepconvpn_u3d_ori

(e) cross_section_of_ground_truth

(f) cross_section_of_pt_u3d_ori

(g) cross_section_of_pt++_u3d_ori

(h) cross_section_of_deepconvpn_u3d_ori

(i) pt_u3d_efl

(j) pt_u3d_acsl

(k) deepconvpn_u3d_efl

(l) deepconvpn_u3d_acsl

(m) cross_section_of_pt_u3d_efl

(n) cross_section_of_pt_u3d_acsl

(o) cross_section_of_deepconvpn_u3d_efl

(p) cross_section_of_deepconvpn_u3d_acsl

Figure 2: Representations of different variants on category building and shrub.

Bibliography

[1] Adams, R., and Bischof, L. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence 16*, 6 (1994), 641–647.

[2] Albitar, C., Graebling, P., and Doignon, C. Robust structured light coding for 3d reconstruction. In *2007 IEEE 11th international conference on computer vision* (2007), IEEE, pp. 1–6.

[3] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 9297–9307.

[4] Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., and Silva, C. T. A survey of surface reconstruction from point clouds. In *Computer graphics forum* (2017), vol. 36, Wiley Online Library, pp. 301–329.

[5] Biffi, L. J., Mitishita, E., Liesenberg, V., Santos, A. A. d., Gonçalves, D. N., Estrabis, N. V., Silva, J. d. A., Osco, L. P., Ramos, A. P. M., Centeno, J. A. S., et al. Atss deep learning-based approach to detect apple fruits. *Remote Sensing 13*, 1 (2020), 54.

[6] Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks 106* (2018), 249–259.

[7] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 11621–11631.

[8] Cai, Y., Chen, X., Zhang, C., Lin, K.-Y., Wang, X., and Li, H. Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 324–333.

[9] Cai, Z., and Vasconcelos, N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence 43*, 5 (2019), 1483–1498.

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (June 2002), 321–357.

[11] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence 40*, 4 (2017), 834–848.

[12] Chen, Y., Zhang, Z., Cao, Y., Wang, L., Lin, S., and Hu, H. Reppoints v2: Verification meets regression for object detection, 2020.

[13] Chen, Z., Zhang, Y., Genova, K., Fanello, S., Bouaziz, S., Häne, C., Du, R., Keskin, C., Funkhouser, T., and Tang, D. Multiresolution deep implicit functions for 3d shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13087–13096.

[14] Cheng, R., Agia, C., Ren, Y., Li, X., and Bingbing, L. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Conference on Robot Learning* (2021), PMLR, pp. 2148–2161.

[15] Chibane, J., Pons-Moll, G., et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems 33* (2020), 21638–21652.

[16] Choe, J., Joung, B., Rameau, F., Park, J., and Kweon, I. S. Deep point cloud reconstruction. *arXiv preprint arXiv:2111.11704* (2021).

[17] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19* (2016), Springer, pp. 424–432.

[18] Crasto, N. Class imbalance in object detection: An experimental diagnosis and study of mitigation strategies, 2024.

[19] Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 9268–9277.

[20] Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Niessner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 5828–5839.

[21] Danielczuk, M., Mousavian, A., Eppner, C., and Fox, D. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021), IEEE, pp. 6010–6017.

[22] Ding, L., and Goshtasby, A. On the canny edge detector. *Pattern recognition 34*, 3 (2001), 721–725.

[23] Dong, Q., Gong, S., and Zhu, X. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1851–1860.

[24] Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N. J., and Wimmer, M. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision* (2020), Springer, pp. 108–124.

[25] Feng, C., Zhong, Y., and Huang, W. Exploring classification equilibrium in long-tailed object detection. In *Proceedings of the IEEE/CVF International conference on computer vision* (2021), pp. 3417–3426.

[26] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F., Fernández, A., García, S., Galar, M., Prati, R. C., et al. Cost-sensitive learning. *Learning from imbalanced data sets* (2018), 63–78.

[27] Fuqua, D., and Razzaghi, T. A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications 150* (2020), 113275.

[28] Gao, W., Zhang, X., Yang, L., and Liu, H. An improved sobel edge detection. In *2010 3rd International conference on computer science and information technology* (2010), vol. 5, IEEE, pp. 67–71.

[29] Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 4857–4866.

[30] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. Recent advances in convolutional neural networks. *Pattern recognition 77* (2018), 354–377.

[31] Guan, H., Zhang, Y., Xian, M., Cheng, H.-D., and Tang, X. Smote-wenn: Solving class imbalance and small sample problems by oversampling and distance scaling. *Applied Intelligence 51* (2021), 1394–1409.

[32] Han, L., Gao, R., Kim, M., Tao, X., Liu, B., and Metaxas, D. Robust conditional gan from uncertainty-aware pairwise comparisons. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 10909–10916.

[33] He, J. Gradient reweighting: Towards imbalanced class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 16668–16677.

[34] He, Y., Yu, H., Yang, Z., Liu, X., Sun, W., and Mian, A. Full point encoding for local feature aggregation in 3-d point clouds. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[35] Hossain, M. S., Betts, J. M., and Paplinski, A. P. Dual focal loss to address class imbalance in semantic segmentation. *Neurocomputing 462* (2021), 69–87.

[36] Huang, J., Artemov, A., Chen, Y., Zhi, S., Xu, K., and Niessner, M. Ssr-2d: Semantic 3d scene reconstruction from 2d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[37] Kamran, S. A., Hossain, K. F., Tavakkoli, A., Baker, S. A., and Zuckerbrod, S. L. Swinvftr: A novel volumetric feature-learning transformer for 3d oct fluid segmentation. *arXiv preprint arXiv:2303.09233* (2023).

[38] Kim, Y. M., Theobalt, C., Diebel, J., Kosecka, J., Miscusik, B., and Thrun, S. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops* (2009), IEEE, pp. 1542–1549.

[39] Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., and Ledoux, H. The hessigheim 3d (h3d) benchmark on semantic segmentation of high-resolution 3d point clouds and textured meshes from uav lidar and multi-view-stereo. *ISPRS Open Journal of Photogrammetry and Remote Sensing 1* (2021), 100001.

[40] Le, D.-V., White, J., Peraire, J., Lim, K. M., and Khoo, B. An implicit immersed boundary method for three-dimensional fluid–membrane interactions. *Journal of computational physics 228*, 22 (2009), 8427–8445.

[41] Le, E.-T., Kokkinos, I., and Mitra, N. J. Going deeper with lean point networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 9503–9512.

[42] Li, B., Yao, Y., Tan, J., Zhang, G., Yu, F., Lu, J., and Luo, Y. Equalized focal loss for dense long-tailed object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 6990–6999.

[43] Li, H., Dong, J., Wen, B., Gao, M., Huang, T., Liu, Y.-H., and Cremers, D. Ddit: Semantic scene completion via deformable deep implicit templates. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 21894–21904.

[44] Li, T., Wen, X., Liu, Y.-S., Su, H., and Han, Z. Learning deep implicit functions for 3d shapes with dynamic code clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12840–12850.

[45] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., and Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems 33* (2020), 21002–21012.

[46] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 2117–2125.

[47] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2980–2988.

[48] Lin, Y., Yan, Z., Huang, H., Du, D., Liu, L., Cui, S., and Han, X. Fpconv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 4293–4302.

[49] Liu, S.-L., Guo, H.-X., Pan, H., Wang, P.-S., Tong, X., and Liu, Y. Deep implicit moving least-squares functions for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 1788–1797.

[50] Liu, Y., Long, W., Shu, Z., Yi, S., and Xin, S. Voxel-based 3d shape segmentation using deep volumetric convolutional neural networks. In *Computer Graphics International Conference* (2022), Springer, pp. 489–500.

[51] Liu, Y., Zhu, K., Wu, G., Ren, Y., Liu, B., Liu, Y., and Shan, J. Mv-deepsdf: Implicit modeling with multi-sweep point clouds for 3d vehicle reconstruction in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 8306–8316.

[52] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 10012–10022.

[53] Lombardi, S., Oswald, M. R., and Pollefeys, M. Scalable point cloud-based reconstruction with local implicit functions. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 997–1007.

[54] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440.

[55] Long, X., Lin, C., Liu, L., Liu, Y., Wang, P., Theobalt, C., Komura, T., and Wang, W. Neuraludf: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 20834–20843.

[56] Ma, B., Liu, Y.-S., Zwicker, M., and Han, Z. Surface reconstruction from point clouds by learning predictive context priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 6326–6337.

[57] McCormac, J., Handa, A., Davison, A., and Leutenegger, S. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)* (2017), IEEE, pp. 4628–4635.

[58] Mehltretter, M. Implicit 3d semantic scene reconstruction.

[59] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 4460–4470.

[60] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106.

[61] Milletari, F., Navab, N., and Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (2016), Ieee, pp. 565–571.

[62] Nguyen, K., Dang, T., and Huber, M. Real-time 3d semantic scene perception for egocentric robots with binocular vision. *arXiv preprint arXiv:2402.11872* (2024).

[63] Nourgaliev, R., Greene, P., Weston, B., Barney, R., Anderson, A., Khairallah, S., and Delplanque, J.-P. High-order fully implicit solver for all-speed fluid dynamics: Ausm ride from nearly incompressible variable-density flows to shock dynamics. *Shock Waves 29* (2019), 651–689.

[64] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999* (2018).

[65] Pan, X., Dai, B., Liu, Z., Loy, C. C., and Luo, P. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844* (2020).

[66] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 165–174.

[67] Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (2020), Springer, pp. 523–540.

[68] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660.

[69] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30* (2017).

[70] Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. *Advances in neural information processing systems 32* (2019).

[71] Ramakrishnan, S. K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A. X., et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238* (2021).

[72] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (2015), Springer, pp. 234–241.

[73] Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R. A., Lucey, S., and Choset, H. Pcrnet: Point cloud registration network using pointnet encoding. *arXiv preprint arXiv:1908.07906* (2019).

[74] Schonberger, J. L., and Frahm, J.-M. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113.

[75] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer soci-*

ety conference on computer vision and pattern recognition (CVPR'06) (2006), vol. 1, IEEE, pp. 519–528.

[76] Singh, V. P., and Frevert, D. K. Watershed modeling. In *World water & environmental resources congress 2003* (2003), pp. 1–37.

[77] Sommer, C., Sang, L., Schubert, D., and Cremers, D. Gradient-sdf: A semi-implicit surface representation for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 6280–6289.

[78] Sulzer, R., Landrieu, L., Marlet, R., and Vallet, B. Scalable surface reconstruction with delaunay-graph neural networks. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 157–167.

[79] Tan, J., Lu, X., Zhang, G., Yin, C., and Li, Q. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 1685–1694.

[80] Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., and Yan, J. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 11662–11671.

[81] Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction 19*, 7 (2010), 829–843.

[82] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[83] Vineet, V., Miksik, O., Lidegaard, M., Niessner, M., Golodetz, S., Prisacariu, V. A., Kähler, O., Murray, D. W., Izadi, S., Pérez, P., et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE international conference on robotics and automation (ICRA)* (2015), IEEE, pp. 75–82.

[84] Wang, C., Deng, C., and Wang, S. Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost. *Pattern Recognition Letters 136* (2020), 190–197.

[85] Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C. C., and Lin, D. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 9695–9704.

[86] Wang, M., Liu, Y.-S., Gao, Y., Shi, K., Fang, Y., and Han, Z. Lp-dif: Learning local pattern-specific deep implicit function for 3d objects and scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 21856–21865.

[87] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and Wang, W. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021).

[88] Wang, T., Zhu, Y., Zhao, C., Zeng, W., Wang, J., and Tang, M. Adaptive class suppression loss for long-tail object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 3103–3112.

[89] Wang, Y., Skorokhodov, I., and Wonka, P. Pet-neus: Positional encoding tri-planes for neural surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12598–12607.

[90] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog) 38*, 5 (2019), 1–12.

[91] Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems 29* (2016).

[92] Wu, S.-C., Tateno, K., Navab, N., and Tombari, F. Scfusion: Real-time incremental scene reconstruction with semantic completion. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 801–810.

[93] Wu, W., Qi, Z., and Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (2019), pp. 9621–9630.

[94] Wunsch, L., Tenorio, C. G., Anding, K., Golomoz, A., and Notni, G. Data fusion of rgb and depth data with image enhancement. *Journal of Imaging 10*, 3 (2024), 73.

[95] Xie, H., Yao, H., Sun, X., Zhou, S., and Zhang, S. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 2690–2698.

[96] Yaman, B., Mahmud, T., and Liu, C.-H. Instance-aware repeat factor sampling for long-tailed object detection, 2023.

[97] Yariv, L., Gu, J., Kasten, Y., and Lipman, Y. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems 34* (2021), 4805–4815.

[98] Yavartanoo, M., Chung, J., Neshatavar, R., and Lee, K. M. 3dias: 3d shape reconstruction with implicit algebraic surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12446–12455.

[99] Yeung, M., Sala, E., Schönlieb, C.-B., and Rundo, L. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics 95* (2022), 102026.

[100] Yun, P., Tai, L., Wang, Y., Liu, C., and Liu, M. Focal loss in 3d object detection. *IEEE Robotics and Automation Letters 4*, 2 (2019), 1263–1270.

[101] Zeng, J., Li, Y., Ran, Y., Li, S., Gao, F., Li, L., He, S., Chen, J., and Ye, Q. Efficient view path planning for autonomous implicit reconstruction. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), IEEE, pp. 4063–4069.

[102] Zhang, S., Li, Z., Yan, S., He, X., and Sun, J. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 2361–2370.

[103] Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 16259–16268.

[104] Zheng, Z., Yu, T., Dai, Q., and Liu, Y. Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 1429–1439.

[105] Zhi, S., Bloesch, M., Leutenegger, S., and Davison, A. J. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 11776–11785.

[106] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 633–641.

[107] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging 39*, 6 (2019), 1856–1867.

[108] Zhou, Z., Zheng, C., Liu, X., Tian, Y., Chen, X., Chen, X., and Dong, Z. A dynamic effective class balanced approach for remote sensing imagery semantic segmentation of imbalanced data. *Remote Sensing 15*, 7 (2023), 1768.

[109] Zhu, S., Wang, G., Blum, H., Liu, J., Song, L., Pollefeys, M., and Wang, H. Sni-slam: Semantic neural implicit slam. *arXiv preprint arXiv:2311.11016* (2023).