S M RIFAIYA ABRAR

Communication-Aware Implicit Neural Fields for Outdoor LiDAR Scene Reconstruction

School of Technology and Innovation

Master's Programme in Computing Sciences

Sustainable and Autonomous Systems

Vaasa 2026

VAASAN YLIOPISTO
School of Technology and Innovation
Author: S M RIFAIYA ABRAR
Thesis title: Communication-Aware Implicit Neural Fields for Outdoor LiDAR Scene Reconstruction
Degree: Master of Science in Technology
Discipline: Sustainable and Autonomous Systems
Supervisor: Jani Boutellier
Instructor: Abol Basher
Year of graduation: 2026 **Number of pages:** 76

ABSTRACT:

The problem of transmission and reconstruction of 3D geometry over wireless channels is an important one in autonomous systems and robotics, where geometry plays an important role, and bandwidth is a critical resource. The current methods for modelling geometry typically represent it as a list of 3D points, which is challenging to maintain continuous surface structure through downsampling to send geometry. The effects are a loss of spatial detail and unevenly reconstructed scenes particularly in outdoor environments with large areas to deal with.

This thesis presents a model that combines implicit neural distance fields with a multi-scale latent representation which is built for wireless transmission. One recent model, called LightNDF, is a lightweight implicit neural field model that reconstructs continuous 3D geometry from voxel occupancy grids using multi-scale CNN features, but was not intended for any kind of transmission function. This work adopts LightNDF as its backbone and significantly extends it for communication-aware deployment. The main contribution is the introduction of a structured latent pyramid which compresses the scene into three spatial pyramids of different resolution. Due to this, the transmission size is decreased from about 218 MB per sample to 0.57 MB. Joint source-channel coding is directly operated in the latent space, and their combination of a bottleneck channel encoder, occupancy-aware masking, a residual cross-scale coding, and an SNR-adaptive gating model, are all beneficial to robustness in the presence of channel noise. Multi-scale features are reconstructed through query based decoding into unsigned distance value at randomized location. Then, outputs are analysed within the SHINE-Mapping pipeline, measuring the spatial consistency between frames.

These experiments are performed on both KITTI and NewerCollege datasets with SNR of 5, 10 and 20 dB, and it can be observed that the proposed model achieves a more stable reconstruction with respect to SEPT, a state-of-the-art wireless point cloud transmission model. Unlike SEPT which achieves better compression by using a single global latent vector, the proposed model maintains the spatial structures across scales and has a better tolerance to channel noise. Results indicate that in the context of mapping and robotics, a spatially structured latent representation is superior to compact single-vector compression despite having a slightly higher transmission cost.

Keywords: point cloud transmission, implicit neural representations, neural unsigned distance fields, joint source-channel coding, Deep Joint Source-Channel Coding, multi-scale latent encoding, wireless 3D reconstruction, semantic communication, incremental mapping, LiDAR, SNR-adaptive transmission, SHINE-Mapping, SEPT, CLLIF

Contents

List of Figures

List of Tables

1 Introduction

The recent advances of 3D capturing technologies like LiDAR and depth cameras (Hassan, Yuan, Mekki, & Chen, 2025; Liu, Liang, Bao, Dong, & Xu, 2025; S. Xie et al., 2024) have made point clouds a widely-used data format for representing the physical world. These include point clouds, representing complex 3D structures, that are crucial in high-precision applications like autonomous driving, robotics and digital twins, where accurate geometric reconstruction of the environment directly affects the performance of the system (Liu et al., 2025). However, when sensors produce millions of points, the sheer size and unstructured nature of point cloud data pose a serious challenge for transmission over bandwidth-limited wireless channels (Shao et al., 2025). When dealing with large scale outdoor scenes, it is not always feasible to send full density point clouds. The main difficulty is to send a sparse semantic 'skeleton' or 'latent representation' to reduce bandwidth (Liu et al., 2025; Shao et al., 2025) and to use neural implicit representations to generate a dense, continuous surface at the receiver (Basher & Boutellier, 2024).

Traditional approaches for point cloud transmission are based on explicit geometry representation and on Separate Source-Channel Coding (SSCC) (Bian et al., 2024; Gündüz, Wigger, Tung, Zhang, & Xiao, 2025; Hassan et al., 2025). Many of them are voxelized or octree-compressed to save memory and data size but come with quantization errors and high memory costs (Shao et al., 2025; Zhao, Jiang, Jia, Torr, & Koltun, 2021). More importantly, these digital schemes suffer from the cliff effect when the channel SNR is below a threshold (Gündüz et al., 2025; Liu et al., 2025).

To address these drawbacks, semantic communication and deep Joint Source Channel Encoding (JSCC) have been developed as a solution that merges compression and channel protection into one end-to-end trainable system, which ensures graceful degradation in the presence of noise as opposed to the sharp cliff effect observed in conventional schemes (Gündüz et al., 2025; H. Xie, Qin, Li, & Juang, 2021). For instance, more recent deep learning-based approaches like SEPT (Bian et al., 2024) that aim to solve the cliff

effect by adopting end-to-end learning in JSCC are not suitable for dense and large point clouds because they face computational complexity and insufficient neighbourhood aggregation. In parallel to this, Implicit neural representations offer a fundamentally different way to encode geometry as continuous learned functions rather than discrete point sets. Lightweight models like LightNDF (Basher & Boutellier, 2024) demonstrate strong reconstruction and densification quality from sparse inputs using this approach. However, they were not designed with any transmission constraints in mind.

The purpose of this thesis is to fill this gap. To achieve this, LightNDF is adopted as the backbone and extended with a structured multi-scale latent pyramid and deep JSCC, transforming it from a reconstruction-only model into a communication-aware system. The objective is to obtain high geometric accuracy and spatial consistency in large scale incremental mapping in realistic wireless channel conditions with a compact transmitted representation for practical deployment.

1.1 Background and Motivation

Modern 3D capturing technologies, such as LiDAR, provide high-precision spatial data, which is critical for autonomous driving, robotics, and large-scale mapping (Shao et al., 2025). In real outdoor scenes like KITTI (Geiger, Lenz, & Urtasun, 2012) and Newer College (Ramezani et al., 2020) dataset, raw point clouds are sparse, unstructured, and increasingly sparse at distance. When it comes to efficient transmission, it's not only about sending fewer points. It is about transmitting a semantic form that allows a receiver to reconstruct a high-fidelity environment from that without transmitting all of the raw points (Liu et al., 2025; Shao et al., 2025). This feature of working with sparse input and producing dense and high-quality reconstruction is significant for real-time communication in resource-limited autonomous systems (Shao et al., 2025).

Traditional transmission techniques, such as SSCC and explicit representations such as voxels or octrees, are widely-used first steps, but suffer from well-known drawbacks (Liu

et al., 2025; Shao et al., 2025). They are not reliable in dynamic outdoor environments due to quantization errors, incomplete coverage in sparse areas, and the cliff effect (Gündüz et al., 2025; Liu et al., 2025). Some of these problems have been tackled by more recent learned approaches, which however introduce additional problems. For instance, SEPT (Bian et al., 2024) encodes the entire point cloud into a single global latent vector, which achieves competitive reconstruction quality under bandwidth-constrained transmission, but the authors point out that the model has limitations in handling large and dense point clouds due to the computational cost of processing many points. In addition to reconstruction quality and robustness of transmission, it is also important to have efficient computation for practical deployment in autonomous systems. Encoders that run on sensor-side hardware must be lightweight enough to process incoming frames in real time, making parameter count and inference speed as important as reconstruction fidelity in communication-constrained settings (Basher & Boutellier, 2024; Safarnejad, Hosein Soheilian, & Safaei, 2025; Shao et al., 2025).

Semantic communication and JSCC offer a solution forward by treating compression and channel protection as a joint problem (Gündüz et al., 2025; Shao et al., 2025). This can be complemented by using implicit neural representations, where learned functions replace discrete geometry and are decoded at any desired resolution (Basher & Boutellier, 2024; Chibane, mir, & Pons-Moll, 2020). The combination of these two directions, end-to-end learned transmission and implicit neural representations, opens up a way towards compact systems, which are very robust against channel noise and can also reconstruct dense geometry from only a small amount of the original data. This is the main motivation of this thesis.

1.2 Problem Statement and Research Questions

1.2.1 Problem Statement

Although there have been recent advances in semantic communication and point cloud compression, combining the strengths of both in a single practical system remains an open challenge. While explicit systems manage to obtain compact representations, they will lose out on continuous surface structure, which can be a significant challenge in downstream applications such as incremental mapping that explicitly rely on geometric precision for navigation quality (Basher & Boutellier, 2024; Wiesmann, Milioto, Chen, Stachniss, & Behley, 2021; Zhong, Pan, Behley, & Stachniss, 2023). On the other hand, implicit neural representations generate high fidelity continuous geometry(Basher & Boutellier, 2024), but were not originally intended to work under transmission constraints, and as such are not directly usable for wireless deployment in their current form. (Gündüz et al., 2025; Liu et al., 2025; Shao et al., 2025).

There is no existing method to connect both sides of this gap. Current approaches do not jointly optimise the implicit geometric reconstruction and channel-aware transmission in a single end-to-end system, and whether such a system can maintain the quality of geometric reconstruction under different channel conditions in a real mapping pipeline remains unknown. In addition to reconstruction quality, the computational complexity of the encoder itself is rarely taken into account in existing works. To be practically deployable, a transmission framework should include a lightweight encoding step that can be performed at the transmitter side on resource-constrained hardware, but this is not the focus of existing approaches.

These gaps form the basis of the research questions and objectives addressed in this thesis.

1.2.2 Research Questions

Based on the identified problem, this thesis aims to answer the following research questions:

- How can implicit neural representations be adapted for communication-aware 3D geometry transmission?
- Can joint source–channel coding be effectively integrated into latent representations to improve robustness under noisy channel conditions?
- How does the proposed approach compare with explicit reconstruction methods such as SEPT in terms of geometric reconstruction quality?
- What is the trade-off between transmission cost and reconstruction fidelity when using latent representations?
- Can a communication-aware 3D reconstruction framework maintain competitive reconstruction quality with significantly fewer parameters than existing models, and is this sufficient for deployment on resource-constrained hardware?
- Do the reconstructed outputs preserve sufficient spatial consistency to support downstream tasks such as incremental mapping?

1.3 Research Objectives

The main objective of this thesis is to create a communication aware framework to reconstruct high-fidelity dense 3D geometry from a wireless transmission over noisy channels. This is done by addressing a series of interlinked objectives.

The first is to embed a structured multi-scale latent representation into an implicit neural field backbone, instead of uncompressed feature grids, which enables a compact

transmission-ready encoding that can greatly reduce the data volume needed for wireless communication. The second is to embed deep JSCC directly in the latent space, which would enable the model to learn representations that are useful regardless of the channel noise and SNR conditions. The third is to verify the surface continuity and spatial information of reconstructed geometry for large scale outdoor scenes. Finally, the framework is used in an incremental mapping pipeline to check the spatial consistency of the reconstructed outputs over the frames under realistic wireless channel conditions.

1.4 Thesis Structure Overview

The remainder of this thesis is organized as follows. Chapter 2 presents a comprehensive literature survey covering point cloud compression, semantic communication, wireless transmission strategies, and reconstruction techniques. Chapter 3 discusses about the comparable frameworks, baselines and models. Chapter 4 describes the proposed methodology, including preprocessing, latent representation design, Joint Source Channel Coding, and integration with incremental mapping. Chapter 5 details the experimental setup and evaluation metrics used to assess the proposed framework and findings, such as analyzing reconstruction quality, transmission efficiency, and mapping performance. Finally, the thesis concludes with Chapter 6 by presenting the limitations and potential directions for future research.

2 Literature Survey

This chapter provides an extensive summary of the state-of-the-art research on point cloud compression, semantic communication, and 3D geometry reconstruction techniques for point cloud transmission. It covers the recent advancements of traditional compression techniques, the introduction of deep-learning based methods (PointNet, transformer based methods), and the paradigm change towards semantic communication via Deep Joint Source–Channel Coding (DJSCC). It also discusses recent progress on reconstruction methods such as explicit point-based decoding and implicit representations, like neural distance fields. This review aims at building a theoretical and methodological basis for the design of communication-aware 3D reconstruction systems. This foundation enables the development of models that efficiently transmit as well as maintain geometric consistency of the models and support downstream tasks like mapping under real-world communication constraints.

The reviewed literature is presented in a methodological structure, which is divided into five sections: (1) Point Cloud Compression, (2) Implicit Neural Representations, (3) Semantic and Deep Joint Source-Channel Communication, (4) Wireless Transmission Strategies, and (5) Reconstruction and Decoding Techniques.

2.1 Point Cloud Compression Techniques

Efficient point cloud compression is the main aspect of 3D data communication and it is directly associated with the transmission latency, memory footprint and quality of reconstruction. The initial approaches were aimed at maximizing compression performance and ensuring geometric fidelity. (Wiesmann et al., 2021) presented a convolutional autoencoder called KPConv that can calculate local descriptors on 3D points without voxelizing them, which enables dense compression of maps. Their approach showed decent rate-distortion performances on datasets such as KITTI and nuScenes, but was primarily designed to be stored offline and not adaptable to dynamic or wireless transmissions.

Designed to work well in the static case, however, it was not made in a way that it can handle changes in bandwidth and packet losses that naturally happen in wireless networks. This is an indication of a weakness in the initial phase of translation of purely geometric compression techniques into real-time semantic transmission systems.

(Que, Lu, & Xu, 2021) proposed VoxelContext-Net which is an octree deep entropy model that is capable of compressing geometry efficiently. Their algorithm performed best on the large datasets, such as MVUB and KITTI, with the help of voxel occupancy probabilities and context modeling between adjacent octree nodes. The hierarchical nature of VoxelContext-Net, which enables fine detail adaptive encoding with low computational costs, enabled high-level feature encoding. Although originally developed for static point clouds, VoxelContext-Net was also evaluated on dynamic point cloud datasets (e.g., 8iVFB), where each frame was treated as an independent static structure. This shows that they were able to achieve competitive compression efficiency without explicit temporal modeling. VoxelContext-Net thus marked a major change towards learned voxel-wise entropy coding, laying the groundwork for subsequent temporal-aware methods in dynamic point cloud compression. Que et al. also mentioned the development of learned compression models such as OctSqueeze, G-PCC, and Learned Point Cloud Compression (LPCC), with the shift towards probabilistic entropy models (based on neural networks) instead of the handcrafted quantization. All of these works represent the trends of increasing integration between geometric structure exploitation and data-driven entropy modeling in point cloud compression.

(Zhao et al., 2021) improved point cloud representation by using the Point Transformer, which employs self-attention in the form of vectors within k-NN neighborhoods. Its hierarchical aggregation of features provided state-of-the-art segmentation and classification. Nevertheless, it was computationally intensive since the local neighborhoods and attention were repeatedly computed, and therefore, it was not easily scalable to real-time communication systems. Moreover, it increased the complexity, thus making it hard to deploy on edge or mobile platforms, especially with strict power and latency requirements. Therefore, even though the Point Transformer architectures provided better ac-

curacy, a trade off was introduced to the model expressiveness and practicality of deployment.

Implicit neural representations are an alternative to explicit point-based encoding and can learn continuous functions over 3D space instead of a compressed vector to represent geometry. (Basher & Boutellier, 2024) proposed a lightweight implicit neural architecture, LightNDF, which encodes geometry in multi-scale feature grids instead of a single compressed vector. Its feature representation, although not conceived as a compression or transmission framework, is structured and can be seen as a semantic encoding of 3D geometry. The architecture and the limitations in the context of transmission are discussed in detail in the following section on implicit neural representations.

2.2 Implicit Neural Representations

Traditional 3D representations like point clouds, voxel grids and meshes have explicit and discrete description of geometry. Although popular, these formats have a drawback; the resolution is determined at the time of representation. If a point cloud has few points, it cannot capture detail in the surface that was not captured in the point, and if a voxel grid has a fixed resolution, such as the 128^3 grid utilized in some embedding networks (Ye, Chen, Wang, & Wang, 2022), it cannot represent surfaces finer than a single voxel due to inherent quantization errors (Shao et al., 2025; Zhao et al., 2021). This makes them unsuitable for applications that have to execute successive surface queries or provide flexible resolution at reconstruction time.

Implicit neural representations go about this in a different way. Unlike traditional representations of geometry as points or grid values, an implicit model encodes a function that maps a query point in 3D space to a scalar value that represents how close the point is to the surface. The most popular methods are occupancy function, which uses whether a query point is inside or outside an object as a prediction target (Mescheder, Oechsle, Niemeyer, Nowozin, & Geiger, 2019) and signed distance function (SDF), which uses the

signed distance to the nearest surface as a prediction target (Park, Florence, Straub, Newcombe, & Lovegrove, 2019). Both representations are continuous by construction, as the model is made of a neural network, which can be sampled at any resolution without retraining. But, both occupancy and signed distance functions require that the surface is watertight, or that the geometry must form a closed volume. This is a severe limitation in real-world LiDAR scans where surfaces are open, partial and frequently incomplete.

In 2020, Chibane et al. (Chibane et al., 2020) proposed the concept of Neural Unsigned Distance Fields (NDF) which substitutes the signed distance with the unsigned distance, without requiring the surface to be closed. This enables the model to simulate thin structures, partial observations and open surfaces which are prevalent in outdoor LiDAR scenes. To encode the entire shape in a single global latent vector, NDF uses multi-scale feature grids extracted by a 3D CNN backbone and sampled through trilinear interpolation at query points and decoded into unsigned distance values. The multi-scale feature extraction enables the model to simultaneously consider both the details of the surface and the structure of the overall shape.

As previously discussed, (Basher & Boutellier, 2024) proposed LightNDF, a lightweight version of NDF that reduces the number of trainable parameters from 4.6 million to 0.46 million while improving reconstruction quality. The LightNDF encoder is a 3D CNN that processes a discrete voxel grid $\mathbf{X} \in \mathbb{R}^{N \times N \times N}$ and produces multi-scale deep feature grids $F_1, \ldots, F_4$ at decreasing spatial resolutions, where earlier grids capture local detail and later grids capture global structure through larger receptive fields. A lightweight decoder consisting of three 1D convolutional layers then takes the features sampled at a query point p from all grids and predicts the unsigned distance to the nearest surface. LightNDF was shown to achieve 2.4 times faster inference than the state-of-the-art while improving point cloud generation quality by up to 24.3% on the ShapeNet Cars dataset.

However, LightNDF was only created as a reconstruction model. The multi-scale feature grids $F_1, \ldots, F_4$ are not structured representations for compression or transmission, but are intermediate CNN activations. This is because there is no restriction on how much

information can be encoded, no method for playing down the dimensions of a spatial or channel feature prior to use, and no discussion about what happens if the features are corrupted or compressed when transmitted over a bandwidth-limited channel. The total data volume of these feature grids at float32 precision is about 216 MB per sample, which is not at all feasible for any kind of wireless communication. This is indeed a gap identified between implicit neural reconstruction and communication-constrained deployment that has not been covered in the literature.

2.3 Semantic and Deep Joint Source–Channel Communication

Semantic-aware JSCC frameworks represent a paradigm shift from the separated source-channel coding, and the conceptual basis of Deep JSCC, which is an end-to-end system where source data is mapped directly to channel symbols, was introduced in (Gündüz et al., 2025). This design avoids explicit error correction and gracefully degrades when the SNR is low and the blocklength is small, as opposed to conventional systems which experience the cliff effect. The cliff effect is one well-known failure mode of separation-based digital communication systems. Even for a slight decline in SNR, the error correction code completely fails and reconstruction quality instantly goes to almost nil when the channel SNR drops below the threshold needed for successful decoding. This is due to the strict separation in traditional systems between source coding and channel coding stages (Bourtsoulatze, Kurka, & Gündüz, 2019). Deep JSCC showed that it was possible to have an adaptive and robust transmission by combining source and channel coding into a single trainable pipeline. The work became the foundation of the later semantic communication methods, indicating the advantages of data-driven joint source-channel optimization.

Expanding on these concepts, (Liu et al., 2025) introduced PCSC, a point cloud semantic communication system based on Voxception-ResNet (VRN) encoders. PCSC combines semantic feature selection with DJSCC to be more robust to noise and allows multiple users using the system by supporting Model Division Multiple Access (MDMA). However,

voxelization, although having its advantages, introduces quantization artifacts and memory overhead, and is not very efficient in preserving geometry at a fine-scale level. This is a limitation that demonstrates there is a trade-off between the Semantic Feature Extraction and the accurate Geometric Reconstruction, indicating the need for improved voxel-based approaches.

A semantic-sensitive classification model built by (Han, Chi, Yang, & Shi, 2023) using Point-BERT is shown to be very accurate at different SNR levels. However, it was mainly concerned with classification and not reconstruction, which demonstrated a necessity to preserve end-to-end geometry in the presence of noisy transmission. According to the study, although semantic extraction might be resilient, it is important to retain the entire geometric structure when using it in real-world applications, including autonomous navigation and AR/VR worlds. Incorporation of both semantic awareness and reconstruction fidelity should therefore be considered in future systems.

(Bian et al., 2024) proposed SEPT (Semantic Point Cloud Transmission), a transformer-based DJSCC system that simultaneously performs semantic encoding, channel adaptation, and reconstruction. The SNR-adaptive self-attention of SEPT enables it to work with a wide range of channel conditions without retraining on each SNR level, avoiding both cliff and leveling effects. In the model, a self-attention block is employed, with noise variance N_0 being concatenated with pooled features and then sent to an MLP to obtain per-channel weights. This end to end design demonstrates the benefits of semantic feature extraction together with powerful channel adaptation for more reliable point cloud transmissions. The suggested SEPT structure, however, is limited in its capability to deal with dense point clouds. Although it works well with small datasets (e.g., ShapeNetV2 $\approx$ 2k points), it performs worse on denser maps (e.g., DePoCo $\approx$ 30k points) due to computational complexity and limited neighborhood aggregation. This presents an important line of future research, which guides the design of future systems that will be able to scale semantic DJSCC frameworks to dense point clouds without compromising the geometric fidelity.

One of the major issues in deep JSCC systems is that the same trained model should be able to cope with different channel conditions without the need of retraining. Initial deep JSCC models (Bourtsoulatze et al., 2019) were trained at a single SNR, in which case the performance drops when the channel condition during inference deviates from that in training. To overcome this, several works have proposed SNR-adaptive mechanism. In the WITT framework, Yang et al. (K. Yang et al., 2023) proposed a Channel ModNet that can accept the current SNR and output the modulation weights to re-scale the latent features of both the encoder and the decoder, enabling the model to flexibly modify its transmission behavior under different channel conditions. As mentioned, SEPT (Bian et al., 2024) developed a similar concept for point cloud transmission using a self-attention block that depends on the noise variance N_0. Both show that conditioning the latent representation on channel state information enhances robustness over SNR, without the need to train distinct models for each SNR. This is especially true in the case of transmitting a structured spatial latent, where the various channels of that latent may have different levels of significance based on the quality of the channel.

2.4 Wireless Transmission and Modulation Strategies

The wireless transmission of point clouds introduces complexities in signal modulation, bandwidth adjustment, and noise resistance. (Shao et al., 2025) introduced a semantic DeepJSCC + Hybrid Digital–Analog (HDA) framework based on NeRF compression, where model parameters are transmitted instead of raw point clouds, enhancing resistance to channel impairments. This method, which directly coded the semantic content into the transmission symbols, mitigates the reliance on the high-fidelity raw data and, therefore, the system is more robust to the SNR variations. Furthermore, the hybrid digital-analog design means graceful degradation as opposed to outright failure, as found in conventional designs.

Classical digital modulation techniques like Binary Phase-Shift Keying (BPSK) with channel coding (1/2 rate) provide reliable communication in moderately high SNR, but suffer from

a fixed block length and latency (Bourtsoulatze et al., 2019). To address these limitations, the authors of (Ibuki, Okamoto, Fujihashi, Koike-Akino, & Watanabe, 2025) proposed a rateless JSCC scheme based on the successive reception channel symbols to generate the point clouds bit-by-bit so as to adaptively change the transmission bandwidth. Zhang et al. in (Zhang et al., 2025) proposed a progressive DJSCC model employing sparse convolutions and Gaussian Mixture Models (GMM) for entropy-based adaptive rate control. These developments point towards convergence of the semantic flexibility and physical-layer efficiency, which is one of the main directions for point cloud communication.

A related area of research concerns hierarchical and residual coding techniques to minimize the amount of redundancy transmitted in representations. Because consecutive frames are highly correlated in classical video coding standards like H.265 (Sullivan, Ohm, Han, & Wiegand, 2012), residual signals between them are sent instead of the content of the frames, as they contain significantly less information than the original frames. This principle was also followed for learned image compression, in which Minnen et al. ((Minnen, Ballé, & Toderici, 2018)) worked on introducing hierarchical priors, meaning one coarse latent representation is used to condition another finer one, thereby lowering the entropy that needs to be communicated at each hierarchical level. It is straightforward to apply these concepts to multi-scale latent representations in 3D scene encoding. If the geometry is represented on spatial grids, and the grid is refined from a coarse grid to a finer grid, the coarse and fine grid representations are by construction correlated. Instead of transmitting each scale individually, this approach transmits only the residual between scales: this means that the variance of the transmitted signal is reduced, giving a direct advantage in robustness against AWGN, as a lower variance signal is less affected by a fixed noise level. This is in contrast to SEPT (Bian et al., 2024) which does not use a hierarchical structure but instead sends a single latent vector across the entire globe, and to standard deep JSCC (Bourtsoulatze et al., 2019) which does not leverage inter-scale correlation but simply applies channel coding to a flat latent. This approach of residual cross-scale coding for implicit neural field latents is a step towards more bandwidth efficient transmission of continuous 3D geometry.

2.5 Reconstruction and Decoding Techniques

The ultimate transmission quality is determined by the precision of the reconstruction. SEPT (Bian et al., 2024) uses a hierarchical decoder based on coordinate reconstruction with progressive upsampling modules, which learns dense point clouds in 3D using compact latent representations. The hierarchical structure of this model enables it to build up fine details of coarse representations, which enhances the overall fidelity of the model. The combination of Point Transformer-based refinement and SNR-adaptive self-attention ensures accurate reconstruction under varying SNR conditions. Such mechanisms illustrate the importance of combining semantic encoding with adaptive reconstruction to maintain quality in noisy channels. Overall, SEPT demonstrates a balance between compact transmission and precise geometry recovery, but only if the point cloud is small and compact.

A similar upsampling method was previously proposed by (Wiesmann et al., 2021), predicting fine-grained structures from compressed latent representations via coordinate and feature expansion layers. This method directly influenced subsequent reconstruction modules in semantic frameworks like SEPT and TSCS. By enabling the expansion of coarse latent codes into dense 3D point clouds, this approach established early principles for mapping compact representations to high-resolution reconstructions, emphasizing the significance of structure-aware upsampling in point cloud transmission. It highlights how latent-space decoding can recover detailed 3D geometry from compact representations.

Models such as TSCS (Hassan et al., 2025) and GenSeC-PC (W. Yang et al., 2026) enhanced reconstruction using cross-attention and diffusion-based denoising, respectively, significantly reducing Chamfer Distance and improving perceptual realism. These generative and attention-based strategies demonstrate that incorporating adaptive mechanisms can substantially improve reconstruction quality compared to purely deterministic approaches. However, their high computational demand limits real-time or edge deployment. Lightweight constructions such as NORRIS (Safarnejad et al., 2025) trade re-

construction detail for reduced complexity, highlighting the persistent trade-off between accuracy, speed, and efficiency. This balance between computational feasibility and reconstruction fidelity remains a key design consideration in practical systems.

From a reconstruction standpoint, LightNDF (Basher & Boutellier, 2024), which is covered in detail in the implicit neural representations section, is also relevant here. Its query-based decoder generates dense point clouds from sparse inputs without the need for explicit meshing by inferring unsigned distance values at various 3D places. With substantially fewer parameters and faster inference, the model reports improvements in Chamfer-based metrics over previous neural distance field baselines. This strong reconstruction behaviour, combined with the structured multi-scale feature representation, makes it a suitable candidate for adaptation toward semantic communication, where the internal representation must serve as a compact and transmission-ready geometric message.

2.6 Conclusion and Research Gap

The literature shows a clear evolution from traditional compression-focused methods toward learning-based approaches for point cloud transmission and reconstruction. Earlier techniques focused on geometric fidelity under stable conditions, but they struggled in dynamic and noisy communication environments. More recent approaches introduced end-to-end learning and improved robustness, yet they often prioritize either compression efficiency or reconstruction quality rather than balancing both. At the same time, there has been a shift toward more flexible representations that better capture continuous geometry, improving reconstruction detail but not directly addressing communication constraints.

Despite these advances, a key limitation remains in the lack of integration between representation, transmission, and downstream usage. Most methods evaluate reconstruction quality in isolation, without considering how outputs behave in practical tasks such as

mapping, where spatial consistency across frames is critical. This highlights a clear research gap: the need for a unified framework that jointly considers compact representation, robustness to communication constraints, and usability in downstream applications. Addressing this gap is essential for enabling reliable 3D perception systems in real-world, communication-limited environments.

3 Baseline Methods and Experimental Framework

3.1 Implicit Comparison Baseline - SEPT

The Semantic Point Cloud Transmission (SEPT) framework (Bian et al., 2024) represents a state-of-the-art approach for point cloud transmission using semantic communication principles. It is designed to jointly perform compression, transmission, and reconstruction of 3D point cloud data in an end-to-end manner. By combining hierarchical feature extraction with transformer-based attention, SEPT is able to capture both local geometric details and global structural information. The model operates directly on point sets, which allows it to process unordered and sparse data effectively. Due to its strong performance in compression-aware reconstruction, SEPT is used in this work as a representative baseline for comparison.

3.1.1 Architecture Overview

SEPT follows an auto-encoder architecture designed for robust transmission under noisy channel conditions. The encoder extracts hierarchical features from input point clouds using PointNet-based Set Abstraction layers and transformer blocks. These components enable the model to progressively reduce the number of points while enriching feature representations with both local and global context. The resulting features are aggregated into a compact latent vector that represents the entire point cloud.

The latent bottleneck incorporates communication-aware design through power normalization and noise injection, simulating transmission over noisy channels. This allows the model to learn representations that are robust to varying signal-to-noise ratio (SNR) conditions. The decoder reconstructs the point cloud through a coarse-to-fine strategy, where an initial set of points is generated and then progressively refined using transformer-based feature enhancement and learned upsampling modules.

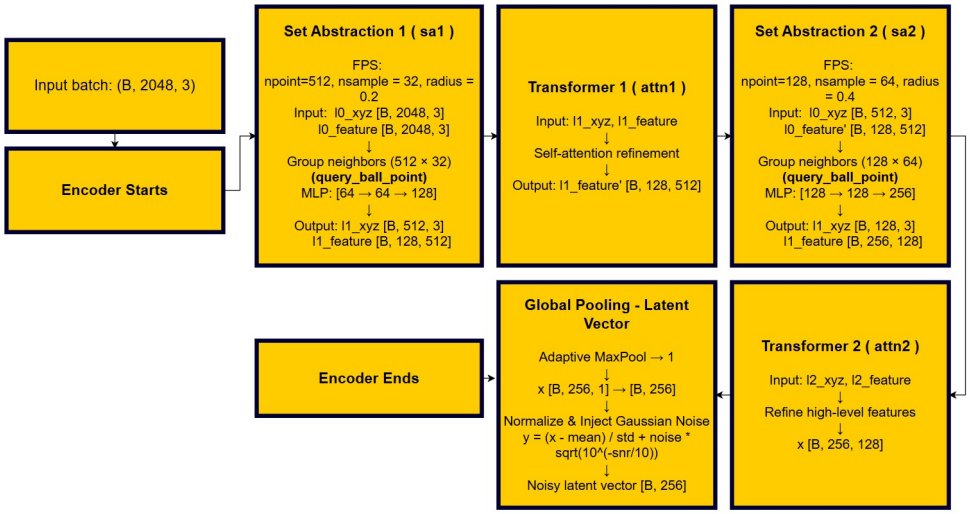


Figure 1. Overview of SEPT's Encoder.

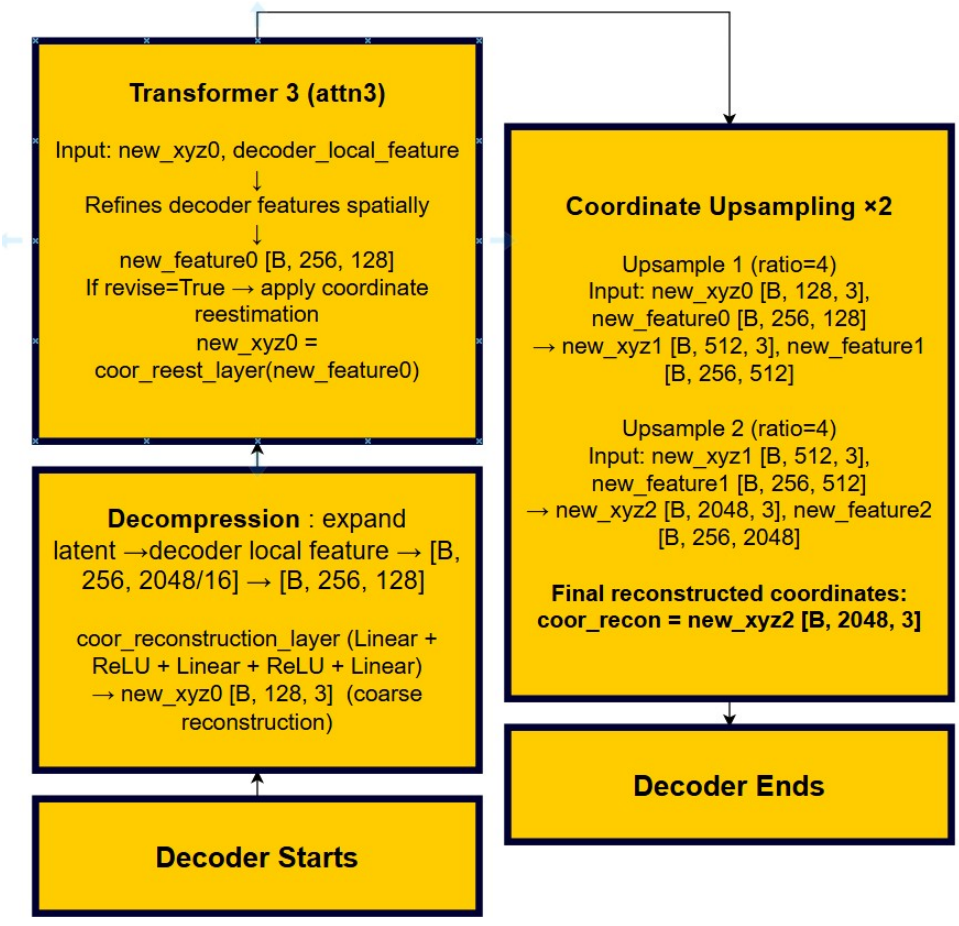


Figure 2. Overview of the SEPT's Decoder.

Method	0 dB	2 dB	5 dB	7 dB	10 dB
SEPT	33.8	34.4	34.9	35.2	35.4
DPCC (Rate 1/2 Polar, 16-QAM)	–	–	–	26.0	35.1
DPCC (Rate 3/4 Polar, QAM)	–	–	33.0	33.0	33.0
G-PCC (Rate 1/2 Polar, 16-QAM)	–	–	27.5	28.5	29.0
DPCC (Rate 1/2 Polar, BPSK)	31.0	29.9	32.3	32.3	32.3
DPCC Upper Bound ($\varepsilon = 10^{-3}$)	32.3	33.5	34.4	35.0	35.4

Table 1. Fig. 1: PSNR D1 reconstruction performance for various SNRs. Table content extracted from (Bian et al., 2024)'s graph.

Method	0 dB	2 dB	5 dB	7 dB	10 dB
SEPT	37.5	38.5	39.0	39.5	40.0
DPCC (Rate 1/2 Polar, 16-QAM)	–	–	–	33.3	39.8
DPCC (Rate 3/4 Polar, QAM)	–	–	38.2	–	–
G-PCC (Rate 1/2 Polar, 16-QAM)	–	–	–	–	38.3
DPCC (Rate 1/2 Polar, BPSK)	34.0	35.1	35.3	35.3	35.3
DPCC Upper Bound ($\varepsilon = 10^{-3}$)	35.3	37.0	38.2	39.6	40.1

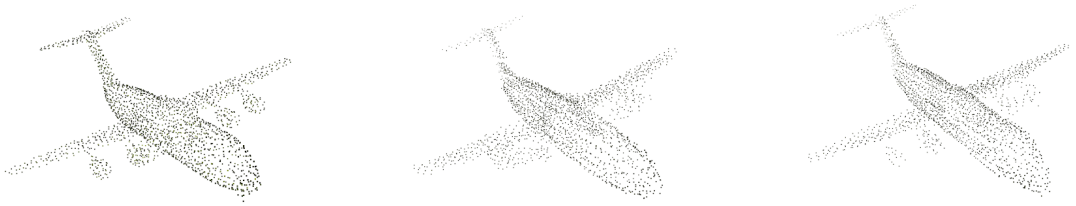
Table 2. Fig. 2: PSNR D2 reconstruction performance for various SNRs. Table content extracted from (Bian et al., 2024)'s graph.

3.1.2 Key Mechanisms

Several key mechanisms contribute to the effectiveness of SEPT:

- **Hierarchical Feature Learning:** Progressive abstraction of point clouds enables efficient encoding of both local and global geometric information.

- **Transformer-based Attention:** Self-attention mechanisms allow modeling of long-range dependencies between points, improving structural understanding.
- **Latent Transmission:** The bottleneck representation is normalized and affected with noise to simulate channel conditions, enabling robustness under varying SNR.
- **Learned Upsampling:** The decoder increases point density through learned offsets, producing finer geometric detail from coarse representations.



(a) Original Point Cloud. **(b)** Newly Trained. **(c)** Reference Reconstruction.

Figure 3. Comparison of Reconstruction between Original Point Clouds, Newly Trained Models Reconstructed Point Clouds and Reconstructed Point Cloud from Author's Best Model.

3.1.3 Limitations in the Context of This Work

Despite its strong compression capability, SEPT reconstructs geometry explicitly as discrete point sets. This representation limits its ability to preserve continuous surface structure and fine geometric consistency, particularly under aggressive compression or noisy channel conditions. Additionally, the model is typically designed for relatively small point sets, which restricts scalability to dense real-world LiDAR data. These limitations become more noticeable in downstream tasks such as mapping, where spatial continuity and consistency across frames are essential.

For these reasons, SEPT serves as a suitable baseline in this thesis to evaluate the trade-off between transmission efficiency and geometric fidelity. The proposed approach aims

to address these limitations by leveraging implicit representations and communication-aware latent modeling.

3.2 LightNDF as a Base Model

LightNDF, proposed by (Basher & Boutellier, 2024), is a neural distance-field-based framework designed for dense point cloud generation from sparse inputs. Unlike explicit reconstruction models, LightNDF represents geometry implicitly as a continuous distance field, allowing surfaces to be reconstructed at arbitrary resolutions. The primary objective of the model is geometric densification rather than compression or communication. Instead of directly predicting point coordinates, it learns a function that encodes the spatial structure of the surface. Due to its strong reconstruction capability and efficient representation of geometry, LightNDF is used in this work as the foundational model for further adaptation.

3.2.1 Architecture Overview

LightNDF follows an encoder–decoder architecture that models geometry as an implicit function. The encoder processes the input, typically represented as a voxelized or discretized form of a point cloud, and extracts multi-scale feature representations that capture both local and global geometric structures. Unlike conventional compression methods, these features are not constrained by a bottleneck or optimized for transmission.

The decoder takes these features together with 3D query points $\mathbf{p} \in \mathbb{R}^3$ and predicts the unsigned distance to the nearest surface. This query-based design allows the model to evaluate geometry continuously across space. By sampling a large number of query points, dense point clouds can be reconstructed without explicitly storing or predicting each point.

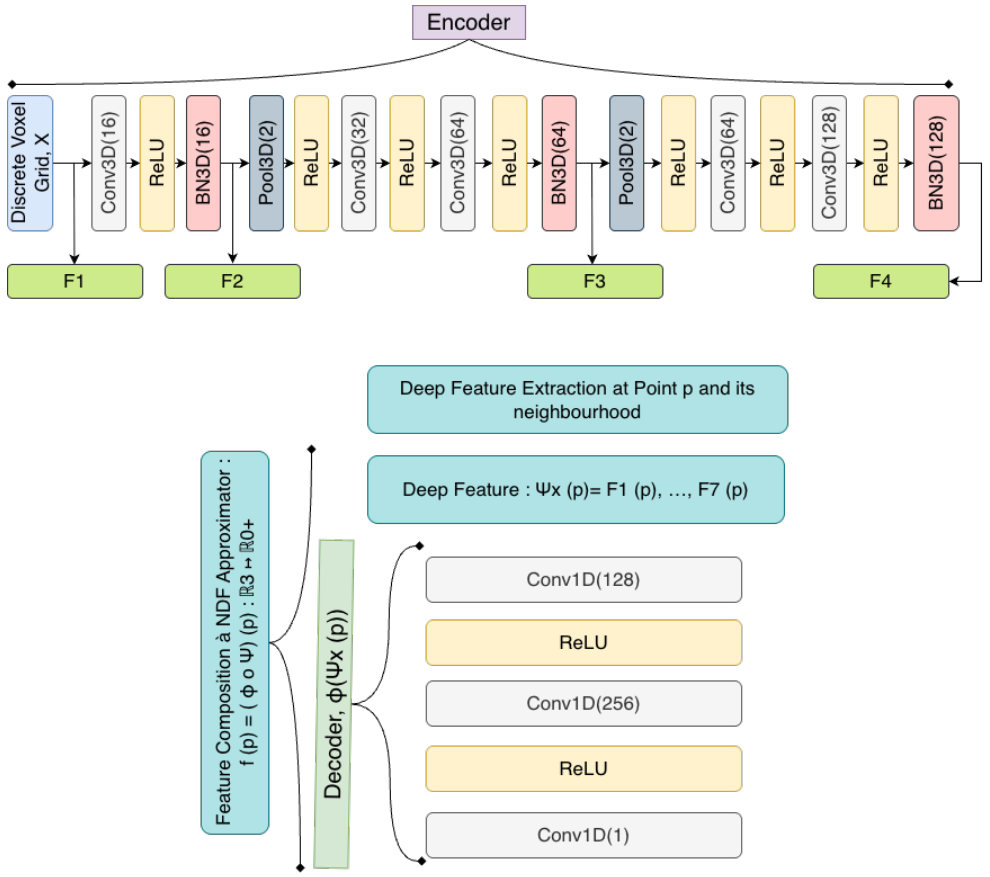


Figure 4. Overview of the LightNDF architecture.

3.2.2 Implicit Reconstruction and Densification

The key characteristic of LightNDF lies in its implicit representation of geometry. Instead of generating a fixed number of points, the model learns a continuous function $f_\theta(\mathbf{p})$ that defines the distance field. The surface is implicitly represented by the zero-level set of this function. As a result, reconstruction quality depends on the number of samples rather than a predefined output size.

This formulation enables the generation of dense and smooth surfaces, even from sparse input observations. Compared to explicit point-based models such as SEPT (Bian et al., 2024), LightNDF provides improved surface continuity and structural consistency. It is particularly effective at inferring missing regions and maintaining geometric coherence across the reconstructed shape.

Method	3000 points (s)	10000 points (s)
NDF (Chibane et al., 2020)	4.63	4.64
GIFS (Ye et al., 2022)	1.66	1.66
LightNDF	**0.69**	**0.68**

Table 3. Inference time comparison for different input sizes. Lower values are better. Table content extracted from (Basher & Boutellier, 2024)'s paper.

Method	# Params	3000 points	10000 points
NDF (Chibane et al., 2020)	4.6M	0.305	0.202
SAL (Atzmon & Lipman, 2020)	4.2M	9.170	7.444
GIFS (Ye et al., 2022)	3.6M	0.330	0.222
LightNDF	**0.46M**	**0.270**	**0.152**

Table 4. Chamfer-L_2 (CD-L_2) comparison ($\times 10^{-4}$). Lower values are better. Table content extracted from (Basher & Boutellier, 2024)'s paper.

3.2.3 Output Characteristics and Evaluation

LightNDF produces a continuous geometric representation rather than a discrete point set. Dense point clouds are obtained by sampling the implicit field at arbitrary resolutions, often resulting in significantly higher point densities than the input. The original evaluation of LightNDF focuses on densification quality, with reconstruction accuracy measured using metrics such as Chamfer Distance.

This approach emphasizes geometric fidelity and flexibility, as the number of output points can be adjusted independently of the model architecture. However, it also means that reconstruction is not inherently constrained by a fixed point budget, which differs from communication-oriented evaluation settings.

3.2.4 Limitations in the Context of This Work

Despite its strong reconstruction capability, LightNDF is not designed for communication-aware scenarios. The internal feature representations are not explicitly constrained in size, and there is no mechanism for controlling transmission cost or ensuring robustness under channel noise. Additionally, the model does not incorporate a structured latent bottleneck, making it unsuitable for direct deployment in bandwidth-constrained environments.

These limitations highlight a key gap between high-quality implicit reconstruction and efficient transmission. In this thesis, LightNDF serves as the base model for developing a communication-aware framework, where its representation is adapted into a compact and robust latent form suitable for transmission while preserving its reconstruction advantages.

3.3 SHINE-Mapping as an Incremental Mapping Baseline

SHINE-Mapping (Zhong et al., 2023) is used in this thesis as an incremental mapping backend for evaluating the spatial usefulness of reconstructed point clouds. Unlike the proposed LightNDF-JSCC model, SHINE-Mapping is not modified in this work. Instead, it receives reconstructed point clouds as input and integrates them into a persistent global map. This allows the evaluation to go beyond frame-level reconstruction and assess whether the reconstructed geometry remains useful for long-term spatial mapping. Therefore, SHINE-Mapping serves as a downstream validation framework for comparing raw point clouds, SEPT reconstructions, and LightNDF-based reconstructions.

3.3.1 Sparse Hierarchical Feature Octree

SHINE-Mapping represents large-scale 3D scenes using a sparse hierarchical implicit structure. The map stores learnable feature vectors at the corners of octree nodes instead

of storing dense voxel values everywhere. This reduces memory usage because only observed regions are allocated. Multiple hierarchy levels allow the map to represent both coarse global structure and fine local detail. In this thesis, the same SHINE-Mapping pipeline is used for all mapping experiments to ensure fair comparison.

3.3.2 Morton Encoding and Hash-Based Storage

To enable efficient map access, SHINE-Mapping uses Morton encoding to convert 3D voxel coordinates into compact one-dimensional keys. These keys are used to index hash tables that store feature vectors at different hierarchy levels. This design allows fast lookup and dynamic expansion of the map as new frames are processed. The use of hash-based storage avoids the need to allocate a dense grid over the full environment. As a result, SHINE-Mapping can scale to longer outdoor sequences such as KITTI and Newer College.

3.3.3 Trilinear Interpolation and Multi-Resolution Feature Fusion

For a query point $\mathbf{x} \in \mathbb{R}^3$, SHINE-Mapping extracts local features by applying trilinear interpolation to the neighboring voxel-corner features. At each hierarchy level h, the interpolated feature is denoted as $F_h(\mathbf{x})$. Features from all hierarchy levels are then combined as

$$F_s(\mathbf{x}) = \sum_{h=0}^{H-1} F_h(\mathbf{x}), \tag{1}$$

where H is the number of hierarchy levels. This multi-resolution fusion enables smooth signed distance prediction at arbitrary spatial locations. It also helps preserve both global consistency and local geometric detail in the reconstructed map.

3.3.4 Online Feature Optimization Using BCE and Eikonal Loss

SHINE-Mapping optimizes local feature vectors using supervision derived from projected signed distances. For a sampled query point x_i, the projected signed distance d_i is mapped to a soft occupancy target using

$$l_i = S(d_i) = \frac{1}{1 + e^{d_i/\sigma}}, \tag{2}$$

where σ controls the softness of the truncation. The predicted signed distance $f_\theta(x_i)$ is mapped similarly as

$$o_i = S(f_\theta(x_i)). \tag{3}$$

The binary cross-entropy loss can be written in the conventional minimization form as

$$\mathcal{L}_{\text{BCE}} = -\left[l_i \log(o_i) + (1 - l_i) \log(1 - o_i)\right]. \tag{4}$$

To encourage valid signed distance behavior, SHINE-Mapping also applies Eikonal regularization:

$$\mathcal{L}_{\text{Eik}} = \lambda_e \left(\|\nabla_x f_\theta(x_i)\|_2 - 1\right)^2. \tag{5}$$

Together, these terms allow the map to learn a smooth and geometrically meaningful implicit surface from sequential point cloud observations.

3.3.5 Regularization Against Catastrophic Forgetting

Since SHINE-Mapping updates the map incrementally, newly observed frames may overwrite previously learned regions. To reduce this issue, the method uses importance-weighted regularization. After optimization for a scan, the importance weight for a parameter θ_i is updated as

$$\Omega_i = \min\left(\Omega_i^* + \sum_{k=1}^{N}\left|\frac{\partial \mathcal{L}_{\text{BCE}}(x_k, l_k)}{\partial \theta_i}\right|, \Omega_m\right), \tag{6}$$

where Ω_m limits the maximum importance value. The regularization term is

$$\mathcal{L}_r = \sum_{i \in A} \Omega_i \left(\theta_i^t - \theta_i^*\right)^2, \tag{7}$$

where A denotes the set of parameters updated in the current iteration. The final incremental mapping loss is therefore

$$\mathcal{L}_{\text{incr}} = \mathcal{L}_{\text{BCE}} + \lambda_e \mathcal{L}_{\text{Eik}} + \lambda_r \mathcal{L}_r. \tag{8}$$

This mechanism is important for maintaining map consistency across long sequences.

3.3.6 Mesh Extraction and Visualization

SHINE-Mapping converts the learned implicit representation into an explicit mesh using marching cubes. This is done by querying the signed distance field over a grid and extracting the zero-level isosurface. In this thesis, the mesh output is used to visually assess the global quality of maps generated from different inputs. These inputs include raw point

clouds, SEPT-reconstructed point clouds, and LightNDF-based reconstructed point clouds. The resulting maps provide qualitative evidence of spatial continuity, surface consistency, and reconstruction artifacts.

3.3.7 Role of SHINE-Mapping in This Thesis

In this work, SHINE-Mapping is used without architectural modification. Its role is to provide a common downstream mapping pipeline for all evaluated reconstruction methods. This allows the thesis to compare not only Chamfer Distance at the point cloud level, but also the mapping usefulness of each reconstruction. The main comparison therefore considers three settings: SHINE-Mapping with raw input point clouds, SHINE-Mapping with SEPT outputs, and SHINE-Mapping with CLLIF outputs. This makes SHINE-Mapping an important evaluation baseline for measuring spatial reconstruction quality beyond isolated frame-level metrics.

4 Methodology

This chapter presents the design and implementation of the proposed framework for communication-aware 3D geometry reconstruction. The proposed model is termed as Communication-aware Latent LightNDF Implicit Fields (CLLIF), where each part of the name reflects a core aspect of the system: the communication-aware channel coding design, the structured multi-scale latent representation introduced for transmission, and the LightNDF implicit neural distance field backbone it extends. The following sections describe each component in detail, covering the encoder backbone and latent pyramid construction, the JSCC channel blocks, the query-based decoder, and finally the incremental mapping pipeline where the reconstructed outputs are evaluated.

4.1 Overview of the Approach

This thesis proposes CLLIF, which is based on the LightNDF model (Basher & Boutellier, 2024) and SHINE-Mapping (Zhong et al., 2023). The objective is not only to get the best geometry reconstruction results after transmission, but also to check if the reconstructed geometry can be used to perform incremental mapping. The end-to-end pipeline takes raw point cloud scans, prepares them for implicit learning, produces multi-scale latent grids from the scans, passes these latent grids through a noisy communication channel, predicts the unsigned distance field from the latent grids, and reconstructs geometry from the distance field prediction and finally passes the reconstructed point clouds for incremental mapping with SHINE-Mapping. This allows one to assess the reconstruction quality and also the downstream mapping quality.

While the base LightNDF (Basher & Boutellier, 2024) was designed exclusively for reconstruction, the proposed model CLLIF includes a structured multi-scale latent bottleneck and incorporates deep JSCC directly into the latent space. This turns the original feature grid into a compact representation that is designed end to end for robust transmission over a noisy wireless channel and geometric reconstruction. It is also different from SEPT

(Bian et al., 2024), because it does not explicitly decode a fixed number of 3D points. Rather, it learns a continuous geometric field and decodes points from the field. This is significant as direct decoding of points (particularly for very compressed representations) can result in inconsistent geometry and/or misplaced points in sparse point clouds. Instead, CLLIF, which uses an unsigned distance field (UDF) for geometry representation, can produce more accurate geometry with smoother surfaces and handle sparse point clouds effectively.

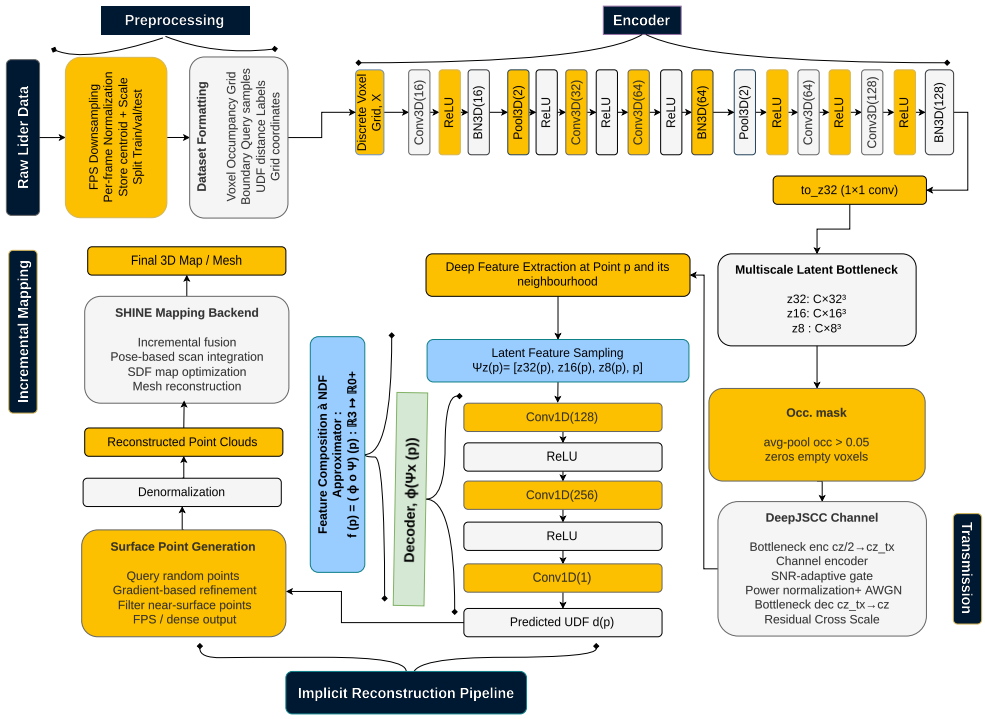


Figure 5. Overview of CLLIF's architecture.

4.2 Data Preprocessing

4.2.1 Point Cloud Downsampling

This research proposes a model that can run with small point clouds and can still generate excellent results. LiDAR scans often have a large number of points, many with varying

densities, so downsampling is used to reduce the point cloud to a more even representation. In this work, Farthest Point Sampling (FPS) is used, which samples points that are well-distributed across the shape. FPS is helpful in this context as it preserves the structure better than random sampling.

Given an original point cloud P and the selected subset S_{k-1}, FPS chooses the next point as

$$p_k = \arg\max_{p \in P} \min_{q \in S_{k-1}} \|p - q\|_2. \tag{9}$$

This means that each new point is chosen as far as possible from the points already selected. As a result, the sampled point cloud covers the scene more uniformly. This is especially important for sparse reconstruction, because missing large parts of the shape during sampling can reduce the quality of the learned distance field.

4.2.2 Coordinate Normalization

Each point cloud is normalized into a cube after downsampling. In this work, the point cloud is mapped into a unit cube $[-1, 1]^3$. Normalization ensures the scale of the input is the same for each scan and each dataset. It also stabilizes the training process as the network always receives the points in the same coordinate space. In the case of the mapping experiments, the centroid and scale factor are saved so that the reconstructed point clouds can be transformed back to to their original real-world coordinates in metres.

4.2.3 Voxelisation and UDF sampling stage

The downsampled LiDAR scans cannot be used directly in the CLLIF approach. The scans have to be converted into a suitable data representation for learning the implicit field.

Therefore, the second stage in the preprocessing is the voxelisation and UDF sampling step. This step transforms each normalized point cloud into a voxel representation for the CLLIF encoder. This involves quantizing the normalized points into a fixed-sized 3D occupancy grid of resolution 128^3. Each voxel is filled with a flag that indicates if the point cloud is present at this voxel. This grid is used as a volume input to the encoder.

The voxelisation and UDF sampling stage also generates training samples around the surface. For each surface point $\mathbf{x}_s$, nearby query points are created by adding Gaussian noise:

$$\mathbf{x}_q = \mathbf{x}_s + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I). \tag{10}$$

The distance to the nearest neighbor of the original surface is then calculated for each sample. The distance is the unsigned distance field. As a result, the voxelisation and sampling stage produces two important things: the occupancy grid (used as input for the encoder), and the UDF supervision values (used for training).

4.2.4 Multi-Scale Boundary Sampling

Boundary samples from various levels of noise are used to train the model. In this adapted LightNDF - CLLIF case, the sampling standard deviations are

$$\sigma = [0.01,\ 0.02,\ 0.04] \tag{11}$$

and the sampling ratios are

$$r = [0.7,\ 0.25,\ 0.05] \tag{12}$$

The smaller standard deviation samples points very close to the surface, which helps improve surface precision. The higher standard deviation samples points further away, which helps the network learn the broader shape structure. The sample ratio prioritizes near-surface samples, since the goal is to accurately recover the surface.

4.2.5 Dataset Preparation Stage

Once voxelisation and sampling are complete, the processed files are rearranged according to CLLIF dataloader requirements. The occupancy grid is saved in a binary format, and boundary samples and UDF targets are saved based on their noise levels. The dataset preparation step also generates the training, validation, and test sets. For mapping experiments, it retains other information such as timestamp, centroid, and scale factor. This allows to transform generated point clouds from the normalized coordinates to the metric coordinates before applying incremental Mapping.

The preprocessing pipeline used in this work follows the methodology of neural implicit function learning. In particular, inspired by Neural Unsigned Distance Fields (Chibane et al., 2020), each point cloud is first converted into a voxelized occupancy grid that serves as the encoder input. To supervise the decoder, boundary samples are generated by affecting surface points with Gaussian noise at multiple scales. The unsigned distance for each sampled point is then computed using nearest-neighbor search, forming the UDF ground truth. This approach is closely related to prior implicit representation methods such as DeepSDF (Park et al., 2019) and Occupancy Networks (Mescheder et al., 2019), which also rely on sampling spatial points and learning continuous geometry representations.

4.3 Feature Engineering

4.3.1 Occupancy Grid Input

The primary feature input to the encoder is a 3D occupancy grid. For a given scan, we voxelize the normalized point cloud to a tensor

$$x \in \mathbb{R}^{B\times 128\times 128\times 128} \tag{13}$$

where B is the batch size. This array is used to convey to the network where geometry is observed in 3D space. The value of the voxel can be 0 or 1 (binary) or stored in a normalized way. The advantages of this representation is that it provides the network with spatial information, which is absent in unstructured points.

4.3.2 Query Points

The second important input is the set of query points. These are 3D locations where the decoder predicts unsigned distance values. Each query point is represented as

$$p \in [-1, 1]^3. \tag{14}$$

In the training phase, query points are sampled from the boundary samples near the surface. During reconstruction, new query points are sampled from the volume to recover surface points. The query-based procedure supports geometry reconstruction at various resolutions without modifying the network structure.

4.3.3 Unsigned Distance Field Targets

The ground-truth value of each query point is the unsigned distance to its nearest surface point. It can be written as

$$d(p, S) = \min_{q \in S} \|p - q\|_2, \tag{15}$$

with S being the surface point set and q the nearest point on the surface. The distance is unsigned so it is always positive. This is helpful for point clouds and open surfaces where inside/outside may not be accurate. So the model learns the distance to the surface at each query point, rather than inside/outside a closed object.

4.4 Model Architecture

4.4.1 Encoder

The encoder takes the occupancy grid and transforms it into learned 3D volumes. It starts with 3D convolutions, which apply small filters that learn to slide over the volume. These filters identify small geometric structures like local occupancy, edges and surfaces. The initial convolution transforms the occupancy input from one channel into more complex features. Subsequent convolutions increase the number of feature channels, giving the network greater capacity to represent complex spatial patterns in the occupancy grid.

A 3D convolution at a point can be thought of as a weighted sum over a local neighbourhood. Applying a convolution filter to nearby voxels produces an output feature that captures local spatial patterns in the occupancy grid. This teaches the model to perceive the local geometry, rather than individual voxels. 3D convolutions are used in this work because the input is a 3D volume. They are the first layer to convert occupancy to

geometry-aware features.

Batch normalization is applied after some convolutions. It normalises the feature map activations to have a stable distribution. Basically, it stops the network from getting stuck with overly large or small activations during training. It makes the model easier to train. This can be particularly helpful for 3D networks because 3D inputs can be large and unstable gradients can cause problems.

ReLU is used after convolutions and normalizations. ReLU is defined as

$$\mathrm{ReLU}(x) = \max(0, x). \tag{16}$$

It leaves positive values unchanged and turns negative values to zero. This introduces non-linearity and enables the network to learn complex surfaces, rather than just linear transformations. Without ReLU, multiple convolution layers won't be expressive enough to represent geometry.

Max pooling is used for reducing spatial resolution. In the encoder, the input resolution is reduced from 128^3 to 64^3, and then to 32^3. Max pooling retains the maximum value in a region, allowing it to preserve features while reducing the amount of computation. It also expands the receptive field, so subsequent features represent large areas of the shape. This makes the encoder more efficient and provides it with more geometric information.

4.4.2 Multi-Scale Latent Grids

Convolution and pooling give the encoder a feature volume of size

$$(B, 128, 32, 32, 32). \tag{17}$$

We then use a $1 \times 1 \times 1$ convolution to project this into a latent grid:

$$z_{32} \in \mathbb{R}^{B \times c_z \times 32 \times 32 \times 32}. \tag{18}$$

In this paper, c_z is the latent channel size. This projection forms a bottleneck as it reduces the number of channels from 128 to a significantly smaller number. The bottleneck is what controls the capacity of the model and has impact on transmission cost.

There are two more downsampling stages that result in latent grids of lower resolution:

$$z_{16} \in \mathbb{R}^{B \times c_z \times 16 \times 16 \times 16}, \tag{19}$$

$$z_8 \in \mathbb{R}^{B \times c_z \times 8 \times 8 \times 8}. \tag{20}$$

The three grids constitute a latent pyramid. The fine grid retains detail, while the coarse grids retain shape information. This is beneficial because geometry reconstruction requires local detail and global consistency.

The lower-resolution latent grids are created using average pooling. Unlike the max pooling operation, which takes the maximum value, average pooling takes the average value. This results in a gradual downsampled feature map and prevents sudden loss of information. In this model, average pooling is used at each downsampling step to produce smooth coarse-to-fine latent grids, where each coarser scale is a stable spatial summary of the finer one. These three latent grids are then passed through the JSCC channel blocks for transmission, as described in the following section.

4.5 Joint Source–Channel Coding in Latent Space

4.5.1 Occupancy-Aware Spatial Masking

LiDAR point clouds are spatially sparse. In a typical scan, the majority of the voxel grid is empty space regardless of the chosen input resolution. Transmitting channel features for empty voxels wastes bandwidth and introduces noise in regions that carry no geometric information. In this work, an input resolution of 128^3 is used, where this sparsity is especially pronounced in large outdoor scenes.

Before the latent grids are passed to the channel encoder, an occupancy mask is computed at each scale. The input voxel grid $x \in \{0,1\}^{B\times 1\times 128^3}$ is average-pooled down to the spatial resolution of each latent:

$$M_s = \mathbf{1}[\mathrm{AvgPool}_s(x) > \tau], \quad s \in \{32, 16, 8\}, \tag{21}$$

where τ is a threshold and $\mathrm{AvgPool}_s$ reduces the grid to resolution s^3. The mask is then applied element-wise:

$$\tilde{z}_s = z_s \odot M_s. \tag{22}$$

This sets latent features in empty regions to zero before encoding. In practice $\tau = 0.05$ is used, which retains any voxel position where at least one occupied cell falls within the pooling window. The mask is not a learned parameter, it is computed directly from the occupancy input at inference time.

4.5.2 Channel Encoder and Decoder

The communication-aware component of the model is the JSCC block applied to each latent grid. Here, a bottleneck is introduced following the principle used in deep JSCC for images (Bourtsoulatze et al., 2019): the channel encoder compresses the latent from c_z channels down to c_{tx} channels before transmission, and the channel decoder expands

back from c_{tx} to c_z at the receiver. In our experiments $c_{tx} = c_z/2$, which halves the number of values transmitted at each scale.

The channel encoder uses a $1\times1\times1$ convolution to reduce channel width, followed by a $3\times3\times3$ convolution that mixes spatial context before the signal leaves the model. The channel decoder mirrors this with a $3\times3\times3$ convolution followed by a $1\times1\times1$ projection back to c_z.

For a masked latent grid $\tilde{z}$ this can be expressed as

$$t = g_\psi(\tilde{z}), \tag{23}$$

where g_ψ is the channel encoder and $t \in \mathbb{R}^{B\times c_{tx}\times s^3}$ is the transmitted representation. After the addition of noise, the channel decoder outputs

$$\hat{z} = h_\phi(\hat{t}), \tag{24}$$

where h_ϕ is the channel decoder. This allows the communication module to be trained jointly with the reconstruction module, so the model learns to preserve vital geometric features under the bandwidth constraint.

4.5.3 Residual Cross-Scale Coding

A residual coding scheme is used across scales, motivated by the hierarchical residual structures used in learned image compression (Minnen et al., 2018) and classical video coding standards such as H.265 (Sullivan et al., 2012). The coarsest grid z_8 is transmitted fully. For the finer grids, only the difference from the upsampled coarser reconstruction is transmitted:

$$r_{16} = z_{16} - \uparrow\hat{z}_8, \tag{25}$$

$$r_{32} = z_{32} - \uparrow\hat{z}_{16}, \tag{26}$$

where $\uparrow$ denotes trilinear upsampling to the next resolution. The channel encoder at each scale operates on the residual rather than the full latent. At the receiver, the latents are reconstructed by adding the upsampled coarser estimate back:

$$\hat{z}_{16} = h_{\phi,16}(\hat{r}_{16}) + \uparrow \hat{z}_8, \tag{27}$$

$$\hat{z}_{32} = h_{\phi,32}(\hat{r}_{32}) + \uparrow \hat{z}_{16}. \tag{28}$$

Because adjacent scales in the pyramid are correlated, the residuals r_{16} and r_{32} have lower variance than the raw latents. A lower-variance signal is less disrupted by a fixed level of AWGN noise, so the effective reconstruction quality at a given SNR improves without any increase in transmitted bandwidth (Minnen et al., 2018; Sullivan et al., 2012).

4.5.4 SNR-Adaptive Channel Gate

A fixed channel encoder sends all c_{tx} channels through the communication link regardless of channel condition. At low SNR, weaker channels are corrupted by noise and contribute little useful information at the receiver, yet they still consume bandwidth.

To address this, a learned gate is applied after the channel encoder, following the idea of content-adaptive rate control in JSCC (K. Yang et al., 2023). A small two-layer MLP takes the current SNR value as input and produces a per-channel sigmoid weight:

$$\gamma = \sigma(W_2 \,\mathrm{ReLU}(W_1 \,\mathrm{SNR}_{\mathrm{dB}} + b_1) + b_2) \in \mathbb{R}^{c_{tx}}, \tag{29}$$

$$t_{\mathrm{gated}} = t \odot \gamma, \tag{30}$$

where the weight γ is broadcast across the spatial dimensions of t. At high SNR, all weights approach one and all channels pass through. At low SNR, the gate suppresses channels that the MLP has learned are less reliable, reducing the effective information rate automatically. The gate is learned end-to-end during training alongside the encoder

and decoder, so no manual rate control is needed.

4.5.5 Power Normalization

The transmitted signal is power-normalized before adding channel noise. This step ensures that the signal has approximately unit average power. Given a transmitted tensor t, the average power is

$$P(t) = \frac{1}{CDHW} \sum_{c,d,h,w} t_{c,d,h,w}^2. \tag{31}$$

The normalized tensor is

$$\tilde{t} = \frac{t}{\sqrt{P(t) + \epsilon}}. \tag{32}$$

This is necessary because otherwise the model could amplify the signal to make the noise less relevant. Power normalization allows the SNR to be meaningful and the channel simulation to be fair. It also makes training more robust to different samples.

4.5.6 AWGN Channel Simulation

Additive white Gaussian noise is used to simulate the channel. For a decibel value of the SNR, the linear SNR is

$$\mathrm{SNR}_{\mathrm{lin}} = 10^{\mathrm{SNR}_{\mathrm{dB}}/10}. \tag{33}$$

With a power-normalized signal, the noise has a standard deviation of

$$\sigma_n = \sqrt{\frac{1}{\mathrm{SNR}_{\mathrm{lin}}}}. \tag{34}$$

The received signal becomes

$$\hat{t} = \tilde{t} + n, \quad n \sim \mathcal{N}(0, \sigma_n^2 I). \tag{35}$$

The model is trained with added noise to the representation to learn how to reconstruct geometry even with a distorted representation. This is why it can be termed communication-

aware. Rather than learning the reconstruction and transmission of a model in isolation, it is learned as a pipeline (Bourtsoulatze et al., 2019).

4.6 Decoder and Geometry Reconstruction

4.6.1 Differentiable Grid Sampling

The decoder reconstructs geometry by querying the transmitted latent grids. For each query point $p \in [-1, 1]^3$, the decoder samples features from z_{32}, z_{16}, and z_8. This is done using differentiable grid sampling, which performs trilinear interpolation at the query location. Instead of selecting only one voxel, it interpolates nearby voxel values to produce a smooth feature vector. This allows the model to decode geometry continuously, even though the latent grids are discrete.

The sampled features are written as

$$f_{32}(p), \quad f_{16}(p), \quad f_8(p). \tag{36}$$

These features contain information from different spatial resolutions. The fine grid supports local detail, while the coarse grids support global structure. This makes the decoder more stable than relying on a single latent scale.

4.6.2 Feature Concatenation

After sampling, features from all scales are concatenated. If coordinate conditioning is used, the query point coordinates are also added to the feature vector. The full decoder input becomes

$$[f_{32}(p), f_{16}(p), f_8(p), p]. \tag{37}$$

Adding coordinates helps the decoder know where the query lies in space. This is useful because two locations may have similar latent features but belong to different parts of the shape. The concatenated feature vector is then passed to the decoder.

4.6.3 UDF Regression with 1D Convolutions

The final decoder is implemented using 1×1 Conv1D layers. These layers work like an MLP applied independently to each query point. The input has shape (B, F, N), where F is the feature dimension and N is the number of query points. The decoder maps each feature vector to one unsigned distance value. The prediction can be written as

$$\hat{d}(p) = \phi([f_{32}(p), f_{16}(p), f_8(p), p]), \tag{38}$$

where ϕ is the MLP-style decoder.

Since the model predicts an unsigned distance field, the output must be non-negative. This is enforced using

$$\hat{d}(p) \leftarrow \max(\hat{d}(p), 0). \tag{39}$$

This prevents negative distance values and keeps the output physically meaningful. The result is a continuous distance field that can be sampled to generate a point cloud.

4.7 Loss Function

The model is trained with mean absolute error (L1) between the predicted and ground truth unsigned distance values. For N query points sampled around the surface, the training loss is

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{d}(p_i) - d(p_i) \right| \tag{40}$$

where $\hat{d}(p_i)$ is the predicted unsigned distance at query point p_i and $d(p_i)$ is the ground truth unsigned distance to the nearest surface point. To avoid large outliers dominating the loss and being unnecessarily weighted, L1 loss is used instead of L2 to obtain sharper surface predictions. The L2 loss penalizes large errors by squaring the loss, causing the model to be more conservative in its predictions of mean values closer to the surface boundary, and therefore making distance fields closer to the boundary more blurry.

The JSCC channel blocks are trained implicitly through $\mathcal{L}$ with no separate supervision. Their weights are fully derived from gradients that are backpropagated through the noise channel, so the model learns to produce a latent representation that will allow them to accurately predict distance even once the representation is corrupted by AWGN. Reconstruction quality is evaluated using Chamfer L2 distance as described in later section.

4.8 Surface Generation

After training, the model generates a reconstructed point cloud by sampling query points and selecting those close to the predicted surface. A surface candidate set can be defined as

$$\mathcal{P}_{\text{surf}} = \{p \mid \hat{d}(p) < \tau\}, \tag{41}$$

where τ is a small threshold. Points with small predicted UDF values are treated as lying near the surface. The final output is then sampled to a fixed number of points, such as 2048 or 4096, for fair comparison. This is important because implicit models can generate many more points than explicit decoders, so evaluation must use the same point budget.

4.9 Integration with SHINE-Mapping

4.9.1 Purpose of Mapping Integration

The reconstructed point clouds are then passed to SHINE-Mapping (Zhong et al., 2023). It is used as a downstream mapping backend to evaluate whether the reconstructed point clouds are spatially consistent enough to produce a coherent map. This is necessary because per-frame metrics, such as Chamfer Distance, measure each frame in isolation and cannot capture whether frames remain consistent when combined together into a full map. Mapping, therefore, acts as a practical test of spatial consistency.

4.9.2 Inverse Normalisation of Reconstructed Points

Before SHINE-Mapping can use the reconstructed points, they must be transformed back from the normalised coordinate space used during training to their original physical scale in metres. The centroid and scale factor stored during preprocessing are used for this step. If $\hat{p}_{norm}$ is a reconstructed normalized point, the metric point can be recovered as

$$\hat{p}_{metric} = \hat{p}_{norm} \cdot s + c, \tag{42}$$

where s is the scale factor and c is the centroid. This step is necessary because SHINE-Mapping uses metric poses and real spatial distances. Without this inverse transforma-

tion, the reconstructed frames would not align correctly in the global map.

4.9.3 Incremental Mapping with Reconstructed Frames

Each reconstructed frame is treated as an observation and passed to SHINE-Mapping together with the corresponding pose information. SHINE-Mapping then updates its sparse hierarchical implicit map frame by frame. Since the same mapping pipeline is used for raw point clouds, SEPT outputs, and CLLIF outputs, the comparison remains fair. The mapping results show whether the reconstructed geometry is consistent enough to support long-term spatial accumulation. This makes SHINE-Mapping a validation step for the practical value of the proposed communication-aware reconstruction model.

4.10 Evaluation Metrics

4.10.1 Chamfer Distance

The geometric difference between the reconstructed point cloud and the ground truth point cloud is measured using the Chamfer Distance. Specifically, the squared (L2) variant is used, where given a reconstructed point set $\hat{P}$ and a ground truth point set P, the Chamfer-L2 distance is

$$\mathrm{CD}(P, \hat{P}) = \frac{1}{|P|} \sum_{p \in P} \min_{\hat{p} \in \hat{P}} \|p - \hat{p}\|_2^2 + \frac{1}{|\hat{P}|} \sum_{\hat{p} \in \hat{P}} \min_{p \in P} \|\hat{p} - p\|_2^2. \tag{43}$$

The first term measures how well the ground truth is covered by the reconstruction. The second term quantifies if the reconstructed points are near the ground truth. The lower the Chamfer Distance the better the reconstruction quality. Please note that the training loss is based on L1 distance field values, whereas this evaluation metric is based on L2 (squared) distance of point positions, which correspond to different quantities: L1 for

the distance field is supervised during training, while L2 is the geometric accuracy of the reconstructed final point cloud. This metric is applied in order to compare SEPT and CLLIF for different data sets and SNR.

4.10.2 Transmission Size

Transmission size is used to measure the communication cost of each method. For SEPT, this refers to the size of the transmitted latent vector of a sample. For the proposed CLLIF method, it refers to the size of the transmitted multi-scale latent grids of a sample after channel encoding and quantization. For the raw baseline, it refers to the size of the original input point cloud. This comparison shows how much data each method needs to send and how that cost relates to reconstruction quality.

4.10.3 Mapping-Based Evaluation

A mapping-based evaluation is performed to consider the usefulness of the reconstructed point clouds in a sequential mapping pipeline. SHINE-Mapping outputs are reviewed for the spatial continuity, structural preservation and completeness of the map. This is particularly significant as frame level metrics cannot always be used to identify if the reconstructed geometry is stable over time. A method which generates less visual artefacts and better accumulated structure is more useful for robotic mapping. Therefore, mapping evaluation complements Chamfer Distance and transmission size for evaluation.

4.11 Summary

To summarize, the methodology starts with processing raw point cloud scans and converting them into a format suitable for implicit reconstruction. FPS downsampling preserves spatial coverage, normalization gives a consistent coordinate range, and the voxelisation and UDF sampling stage creates both occupancy inputs and UDF supervision

targets. Next, the proposed CLLIF model encodes the occupancy grid into multi-scale latent grids, and directly implements JSCC in the latent space. After power normalization and AWGN channel simulation, the decoder reconstructs geometry through grid sampling and UDF regression. Finally, the reconstructed point clouds are evaluated using Chamfer Distance, transmission size and SHINE-Mapping to assess reconstruction quality and mapping usefulness.

5 Experiments, Results, and Findings

This chapter describes the experimental setup, results, and findings of this work. The experiments evaluate the proposed CLLIF model against SEPT under realistic wireless channel conditions across two outdoor LiDAR datasets.

5.1 Dataset Setup

Two datasets are used in this work: KITTI (Geiger et al., 2012) and Newer College (Ramezani et al., 2020), covering outdoor driving scenes and dense real-world campus scans respectively. Each frame in both datasets is first processed and downsampled using farthest-point sampling. For SEPT, the point clouds are reduced to 4096 points per frame, since the model expects a fixed-size point input. For the proposed CLLIF model, the same datasets are further processed into occupancy grids and UDF supervision as described in the methodology.

For KITTI, the model is trained on 500 samples, validated on 100, and tested on 100 samples. For Newer College, the training set consists of 8677 samples, with 950 for validation and 1300 for testing. This gives a mix of scene types and densities across both datasets.

5.2 Experimental Setup

Both models are trained from scratch on the datasets described above. For the proposed CLLIF model, training is conducted using the Adam optimiser with a learning rate of 1×10^{-4} over 50 epochs with a batch size of 2. The training loss is mean absolute error (L1) between the predicted and ground truth unsigned distance values, computed over 200,000 query points per sample. Query points are sampled at multiple distances from the surface following sigma values of $[0.01, 0.02, 0.04]$ with a ratio of $[0.7, 0.25, 0.05]$, ensuring the model learns both fine surface detail and broader shape structure. SEPT is

trained following its original configuration as described in (Bian et al., 2024).

The communication channel is simulated as an AWGN channel throughout training and evaluation. Gaussian noise is injected directly into the latent representation during training with a fixed SNR of 10 dB, and the JSCC channel blocks are trained end-to-end with the geometric reconstruction. This implies that the model can learn to encode geometry in such a way that it is robust to channel noise without requiring a dedicated channel coding step. To evaluate, experiments are carried out at SNR = 5 dB, 10 dB, and 20 dB to examine the performance under various channel conditions. For direct comparison, SEPT is evaluated using the same SNR values.

Another important point is that the proposed model reconstructs geometry through an implicit distance field. This makes it possible to densify the point cloud during the reconstruction. It can, therefore, generate smoother and more complete surfaces even when provided with sparse inputs, as opposed to explicit point-based decoding.

5.3 Evaluation Metrics

Two main types of evaluation are used in this work.

First is Chamfer Distance, which measures how close the reconstructed point cloud is to the ground truth. Lower values mean better reconstruction quality. Second is transmission size. This measures how much data needs to be sent. Three cases are compared:

- Raw point cloud (baseline)
- SEPT latent representation
- Proposed LightNDF latent grids

In addition, mapping-based evaluation is used. The reconstructed frames are fed into

SHINE-Mapping to see how well they support incremental map building.

5.4 Reconstruction Quality Results

The first set of results compares reconstruction quality using L2 Chamfer Distance.

Table 5. Chamfer Distance Comparison ($\times 10^{-4}$) with 4096 Points (Lower is Better).

Method	KITTI	Newer College
SEPT	10.0	32.0
CLLIF (Proposed)	**3.4**	**4.6**

From this table, the difference is quite clear. SEPT performs well on smaller or structured datasets, but its performance drops when dealing with larger, real-world scenes. The proposed model improves the Chamfer Distance significantly. And this happens even without densification, a core feature of this model.

5.5 Transmission Size Comparison

The next comparison focuses on how much data each method needs to transmit.

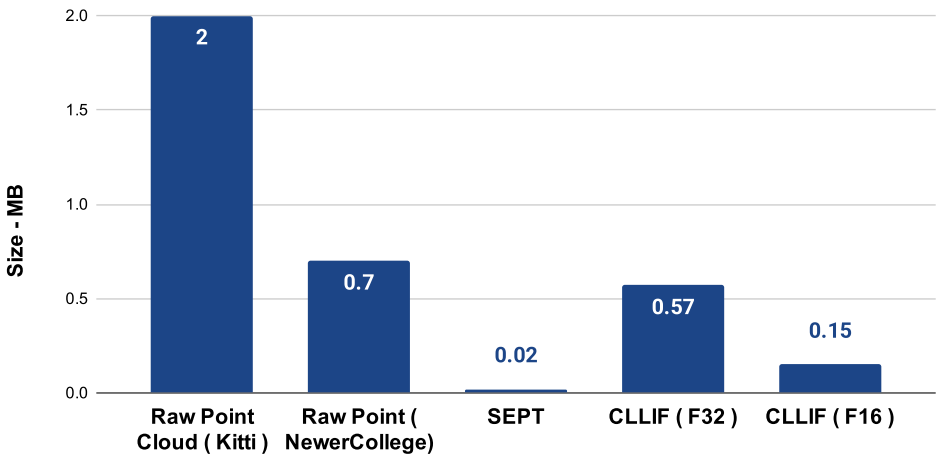


Figure 6. Transmission Size Comparison.

SEPT clearly has the smallest latent size. That is expected, since it compresses the entire point cloud into a single vector.

Unlike SEPT, which compresses point clouds into a compact global latent representation, CLLIF encodes geometry as a multi-scale implicit function using spatial latent grids. This representation inherently requires more storage but enables continuous surface reconstruction, improved geometric fidelity, and robustness to noise. Therefore, LightNDF prioritizes representation quality over extreme compression, while still allowing moderate compression through quantization techniques.

Our proposed CLLIF with latent is much more practical then Base LightNDF. It is almost 99.74% smaller.

Table 6. LightNDF Size Comparison.

Representation	**Size (KB)**	**Relative**
Base LightNDF	216 MB	100%
CLLIF (F32) Latent (Proposed)	0.57 MB	0.26%
CLLIF F(16) Latent (Proposed)	**0.15 MB**	**0.07%**

So the takeaway is simple: SEPT is more compact, but CLLIF is more informative.

5.6 Model Complexity and Parameter Size

Another key comparison is the size of the models themselves.

Table 7. Model Parameter Size Comparison.

Model	**Parameters (Millions)**	**Relative Size**
SEPT	21.7 M	100%
LightNDF	0.46 M	2.3%
CLLIF (Proposed)	**0.475 M**	2.3%

This is one of the strengths of the proposed model. Even though it uses volumetric input

and multi-scale latents, the overall parameter count is much smaller than SEPT.

A smaller model has a few advantages:

- Faster inference time
- Lower memory usage
- Easier deployment on edge systems

So while the proposed model uses a slightly larger latent representation for transmission, the model itself is lightweight. It introduces channel encoding, residual cross-scale coding, and an SNR-adaptive gate on top of the LightNDF backbone, yet the total parameter count increases by only approximately 15,000 parameters, from 0.46 million to 0.475 million. This means the communication-aware components add less than 4% overhead to the original model size, keeping the system well within the lightweight category suitable for deployment on resource-constrained hardware. This makes it practical for real-world systems where both computation and communication matter.

5.7 SNR Sensitivity Analysis

To understand how robust each model is to noise, experiments are run at different SNR levels.

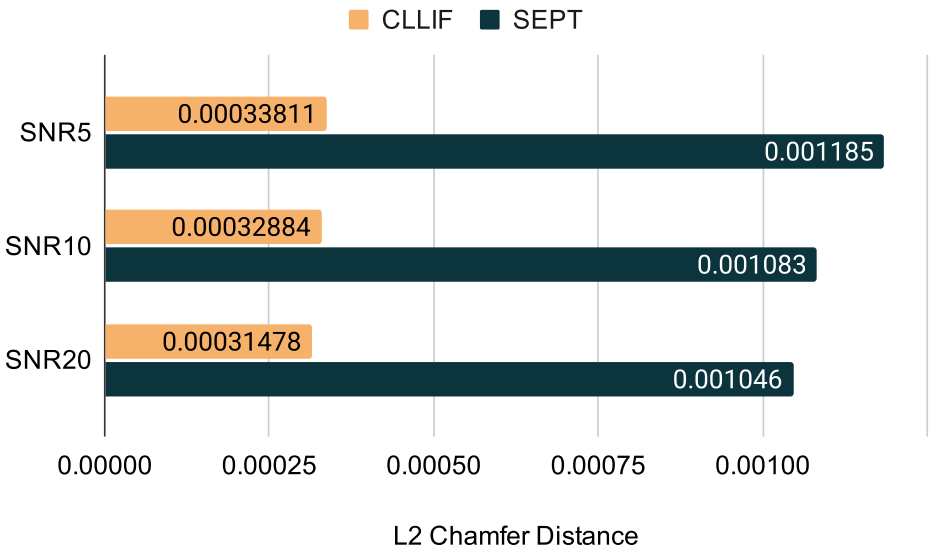


Figure 7. Chamfer Distance Comparision between Proposed CLLIF and SEPT at SNR5, SNR10 and SNR20 (Lower is Better) .

Both models improve as SNR increases, as expected. However, across experiments on the KITTI dataset at SNR values of 5, 10, and 20 dB, the proposed CLLIF model shows more stable reconstruction quality at lower SNR compared to SEPT (Bian et al., 2024).

SEPT compresses the entire point cloud into a single global latent vector via max pooling before transmission. Any noise on that vector affects the entire reconstruction, since there is no spatial structure left at the receiver to recover from. The proposed model instead transmits a spatial latent pyramid of three grids z_{32}, z_{16}, and z_8 as residual signals. Even if noise corrupts the fine-scale residual r_{32}, the coarse shape in z_8 remains intact and the decoder can still produce a geometrically consistent output.

The SNR-adaptive gate further contributes to this stability. At low SNR, the gate suppresses weaker channels via a learned per-channel sigmoid weight $\gamma \in \mathbb{R}^{c_{tx}}$, reducing the noise impact without discarding the spatial pyramid structure. SEPT's SA block conditions on the noise variance N_0 to modulate its single latent vector, but has no equivalent fallback when that vector is heavily corrupted. The combination of residual pyramid

coding and the SNR-adaptive gate is what keeps reconstruction quality more consistent across channel conditions.

5.8 Qualitative Reconstruction Results

(a) Original Point Cloud. **(b)** CLLIF . **(c)** SEPT.

Figure 8. Reconstruction comparison at SNR = 10 dB and 4096 points per sample for Newer College datasets.

(a) Original Point Cloud. **(b)** CLLIF. **(c)** SEPT.

Figure 9. Reconstruction comparison at SNR = 10 dB and 4096 points per sample for Kitti datasets.

Visually, the difference is noticeable. SEPT outputs tend to look more scattered, especially in complex regions. they also tend to densify in the starting region. The proposed model produces smoother surfaces and better preserves structure. This is consistent with the Chamfer Distance results.

5.9 Point Cloud Densification Capability

A key advantage of the proposed CLLIF model is its ability to densify point clouds during reconstruction. This is because this model learns a continuous unsigned distance field, rather than explicit models like SEPT, which output a fixed set of output points. This means that the number of output points is not determined by the network architecture. Rather, it is based on the number of query points sampled during decoding. This provides flexibility to the model to adapt the reconstruction density without retraining.

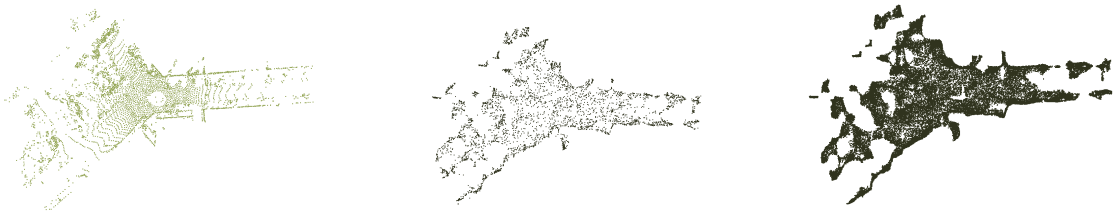
In practice, it implies that the same trained model is able to generate sparse and dense output. By increasing the number of sampled query points, more surface points can be extracted from the learned geometry. This is particularly useful for large outdoor datasets such as KITTI and Newer College, where sparse outputs are likely to miss out on fine details of structures. The higher the sampling, the more complete and consistent surfaces are created. As a result, the model can better represent complex real-world environments.

(a) Original Point Cloud. **(b)** CLLIF. **(c)** Densed 60k points.

Figure 10. Densification results on Newer College dataset, showing improved surface continuity and detail.

The impact of densification becomes clearer when comparing visual results. Sparse reconstructions often leave gaps between points, especially in regions with curved or irregular surfaces. After densification, these gaps are reduced, and surfaces appear smoother and more continuous. This is noticeable in areas such as roads, building walls, and sur-

rounding structures. The overall geometry becomes easier to interpret and closer to the original scene.



(a) Original Point Cloud. **(b)** CLLIF. **(c)** Densed 60k points.

Figure 11. Densification results on KITTI dataset. From left to right: sparse input (4096 points), CLLIF reconstruction, and densified CLLIF output.

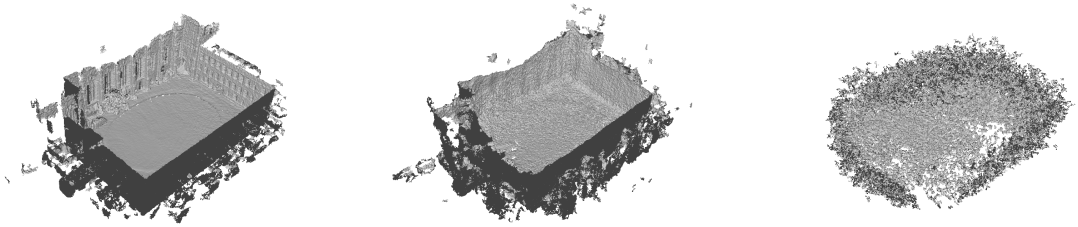
This capability also has a direct effect on incremental mapping. SHINE-Mapping relies on consistent geometric observations across frames to build a stable global map. When sparse or noisy reconstructions are used, the mapping process may struggle to maintain continuity. Densified outputs provide stronger and more consistent geometric signals, which improves interpolation and alignment. This leads to smoother and more complete maps over time.

Another important benefit is flexibility at inference time. In explicit models like SEPT, increasing the number of output points usually requires modifying the decoder or retraining the model. In contrast, the proposed approach allows this adjustment simply by changing the number of query samples. This makes it easier to balance between transmission cost and reconstruction quality depending on the application. It also makes the model more adaptable to different deployment scenarios.

Overall, the densification capability shows that the proposed model is more than just a reconstruction method. It provides a flexible representation that can scale its output quality when needed. This is particularly useful for mapping, visualization, and downstream tasks that require dense geometry. This makes it a strong candidate for real-world

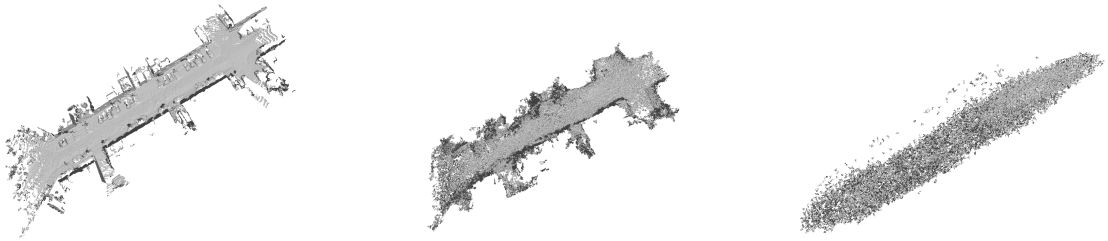
systems where both quality and adaptability matter.

5.10 Mapping Based Evaluation with SHINE-Mapping



(a) Original Point Cloud. **(b)** CLLIF. **(c)** SEPT.

Figure 12. SHINE-Mapping results of 1300 Frames. Comparison between raw input, SEPT reconstruction, and CLIIF reconstruction for Newer College dataset..



(a) Original Point Cloud. **(b)** CLLIF. **(c)** SEPT.

Figure 13. SHINE-Mapping results of 100 frames. Comparison between raw input, SEPT reconstruction, and CLLIF reconstruction for KITTI dataset.

Table 8. L2 Chamfer comparison of SHINE-Mapping Between CLLIF vs SEPT.

Representation	L2 Chamfer (NewerCollege)	L2 Chamfer (Kitti)
CLLIF	**0.34**	**0.27**
SEPT	1.05	0.89

The mapping results highlight something important. Even when SEPT produces a compact representation, the reconstructed frames are not always stable enough for mapping. Over time, this leads to inconsistencies in the map.

The proposed model performs better here. The reconstructed frames align more consistently across time. The resulting maps look more continuous and complete. This shows that reconstruction quality alone is not enough. The geometry also needs to be stable for downstream tasks.

6 Conclusion and Limitation

6.1 Conclusion

This thesis aimed to determine if an implicit neural field designed for reconstruction quality could be adapted to become a usable wireless transmission system without compromising the attributes that make it useful. The results indicate that it can. The findings emphasize two general themes relating to the representation and transmission of geometry in wireless reconstruction systems. The LightNDF model was designed purely for reconstruction quality and densification, with no consideration for transmission. There was no latent bottleneck, no channel model, and no concept of how much data would actually need to move over a wireless link. CLLIF made a fundamental change by introducing the multi-scale latent pyramid and the JSCC channel blocks, transforming it into a system that can operate over a real channel.

One of the most concrete results is the reduction in transmission size. The LightNDF would have to send the entire feature volume, 216 Mb per sample at float32. CLLIF's latent representation reduces this to about 0.57 MB per sample, or approximately 380 times less. This was done by encoding the backbone features into small latent grids at three spatial scales, adding a bottleneck channel encoder that halves the channel width before sending, and by setting the voxels in empty space to zero with the occupancy mask. The geometry is still preserved across scales, so the decoder receives enough information to reconstruct accurate point clouds despite the much smaller transmitted payload.

The comparison with SEPT is useful because it represents a different design philosophy rather than a direct competitor. SEPT is designed with a focus on transmission from the ground up, and its compression is very high due to the fact that it collapses the point cloud into one global vector. This is suitable for compact synthetic objects, like the ones SEPT was designed and tested with. The proposed model works in a different space, from an implicit neural representation initially designed for densification, which also works quite

well with large outdoor scenes, and adapting it for wireless transmission without losing the spatial structure. The two models are addressing similar, but not identical, problems, and that is what the results show.

The noise robustness results further elaborate on this. At SNR 5dB on KITTI, the proposed CLLIF model degrades more gradually than SEPT, which connects back to the structure of the latent pyramid. The coarse grid preserves the global shape and when fine-scale residuals are corrupted by noise, the coarse grid can be used as a fallback. The SNR adaptive gate also helps this, by lowering the rate of the transmission for weaker channels when the SNR is low, instead of sending them noisy at full rate. LightNDF did not have these properties, and they are important because the model is being applied in a transmission context with variable channel conditions.

Overall, the findings show that the adaptation of an implicit neural field for wireless transmission is possible and beneficial. The difference between 216 MB and 0.57 MB per sample demonstrates that the latent design does most of the work and the channel-aware components are responsible for noise and varying SNR. For applications like outdoor mapping and robotics where geometric accuracy matters and scenes are large and sparse, this kind of approach produces more consistent results than compressing everything into a single vector, while still being practical to transmit over a bandwidth-limited channel.

6.2 Limitation

There are a number of challenges with the proposed system that indicate areas that need further development. Even though the transmission size is much smaller than LightNDF, it is still larger than compression methods designed specifically for this purpose, such as SEPT. This indicates that even with advantages obtained by the bottleneck encoder and occupancy masking, the suggested model is at a disadvantage in extremely bandwidth-constrained conditions. Moreover, Throughout the work, the channel model employed is AWGN, commonly assumed in the literature of deep JSCC and which does not fully re-

flect the circumstances of practical wireless channels, where fading and interference are present. These factors would affect the received latent quality in ways the current model was not trained to handle. Finally, the preprocessing pipeline requires offline computation of voxel grids and UDF supervision before training can begin, which limits the model from being applied to new scenes without prior data preparation.

Future research should focus on ways to reduce the latent size even further while preserving the spatial structure of the proposed model that enables it to work in large outdoor environments. A smaller latent representation that also captures multi-scale geometry may reduce the transmission cost closer to what has been found with the single-vector methods (e.g., SEPT) without sacrificing the reconstruction quality and noise robustness demonstrated in this work. Together, these methods can lead to a more reliable system with more ease of transmission in the real world.

7 Acknowledgements

The author would like to acknowledge the use of AI-assisted tools during the preparation of this thesis. Large language models including ChatGPT and Claude were used to support tasks such as code suggestions, technical discussion, and sentence refinement. Quillbot and Grammarly were used for grammar checking and sentence enhancement. All technical content, results, analysis, and conclusions are the sole work of the author. These tools were used only to assist with language and presentation, not to generate or validate any of the research findings.

Bibliography

Atzmon, M., & Lipman, Y. (2020, June). SAL: Sign Agnostic Learning of Shapes From Raw Data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2562–2571). https://doi.org/10.1109/CVPR42600.2020.00264

Basher, A., & Boutellier, J. (2024). Convolutional Neural Network-Based Efficient Dense Point Cloud Generation Using Unsigned Distance Fields. In X.-S. Yang, S. Sherratt, N. Dey, & A. Joshi (Eds.), *Proceedings of Ninth International Congress on Information and Communication Technology* (pp. 507–515). Singapore: Springer Nature.

Bian, C., Shao, Y., & Gündüz, D. (2024, September). Wireless Point Cloud Transmission. In *2024 IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (pp. 851–855). https://doi.org/10.1109/SPAWC60668.2024.10694621

Bourtsoulatze, E., Kurka, D. B., & Gündüz, D. (2019, May). Deep Joint Source-channel Coding for Wireless Image Transmission. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774–4778). https://doi.org/10.1109/ICASSP.2019.8683463

Chibane, J., mir, M. A., & Pons-Moll, G. (2020). Neural Unsigned Distance Fields for Implicit Function Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 21638–21652). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/f69e505b08403ad2298b9f262659929a-Paper.pdf

Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354–3361). https://doi.org/10.1109/CVPR.2012.6248074

Gündüz, D., Wigger, M. A., Tung, T.-Y., Zhang, P., & Xiao, Y. (2025, September). Joint Source–Channel Coding: Fundamentals and Recent Progress in Practical Designs. *Proceedings of the IEEE*, *113*(9), 888–919. https://doi.org/10.1109/JPROC.2024.3477331

Han, T., Chi, K., Yang, Q., & Shi, Z. (2023, December). Semantic-aware Transmission for Robust Point Cloud Classification. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference* (pp. 7617–7622). https://doi.org/10.1109/GLOBECOM54140.2023.10437861

Hassan, M. M., Yuan, H., Mekki, H. A. S., & Chen, Z. (2025, May). Transformer-Based Semantic Communication System for 3D Point Cloud Transmission. In *2025 IEEE International Workshop on Radio Frequency and Antenna Technologies (iWRF&AT)* (pp. 496–500). https://doi.org/10.1109/iWRFAT65352.2025.11102914

Ibuki, S., Okamoto, T., Fujihashi, T., Koike-Akino, T., & Watanabe, T. (2025). Rateless Deep Joint Source Channel Coding for 3D Point Cloud. *IEEE Access*, *13*, 39585–39599. https://doi.org/10.1109/ACCESS.2025.3546514

Liu, X., Liang, H., Bao, Z., Dong, C., & Xu, X. (2025, January). A Semantic Communication System for Point Cloud. *IEEE Transactions on Vehicular Technology*, *74*(1), 894–910. https://doi.org/10.1109/TVT.2024.3456099

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019, June). Occupancy Networks: Learning 3D Reconstruction in Function Space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4455–4465). https://doi.org/10.1109/CVPR.2019.00459

Minnen, D., Ballé, J., & Toderici, G. D. (2018). Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In *Advances in Neural Information Processing Systems* (Vol. 31, pp. 10771–10780). Curran Associates, Inc. Retrieved from [2026-06-03]https://proceedings.neurips.cc/paper/2018/hash/53edebc543333dfbf7c5933af792c9c4-Abstract.html

Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019, June). DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 165–174). https://doi.org/10.1109/CVPR.2019.00025

Que, Z., Lu, G., & Xu, D. (2021, June). VoxelContext-Net: An Octree based Framework for Point Cloud Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6038–6047). https://doi.org/10.1109/CVPR46437.2021.00598

Ramezani, M., Wang, Y., Camurri, M., Wisth, D., Mattamala, M., & Fallon, M. (2020, October). The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4353–4360). https://doi.org/10.1109/IROS45743.2020.9340849

Safarnejad, S., Hosein Soheilian, M., & Safaei, B. (2025, July). NORRIS: Noise-Resilient and Resource-Efficient Semantic Encoded Point Cloud Data Transmission for Internet of Things Communications. *IEEE Internet of Things Journal*, *12*(13), 25720–25731. https://doi.org/10.1109/JIOT.2025.3559435

Shao, Y., Bian, C., Yang, L., Yang, Q., Zhang, Z., & Gündüz, D. (2025, December). Point Cloud in the Air. *IEEE Communications Magazine*, *63*(12), 142–148. https://doi.org/10.1109/MCOM.001.2400541

Sullivan, G. J., Ohm, J.-R., Han, W.-J., & Wiegand, T. (2012, December). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(12), 1649–1668. https://doi.org/10.1109/TCSVT.2012.2221191

Wiesmann, L., Milioto, A., Chen, X., Stachniss, C., & Behley, J. (2021, April). Deep Compression for Dense Point Cloud Maps. *IEEE Robotics and Automation Letters*, *6*(2), 2060–2067. https://doi.org/10.1109/LRA.2021.3059633

Xie, H., Qin, Z., Li, G. Y., & Juang, B.-H. (2021). Deep Learning Enabled Semantic Communication Systems. *IEEE Transactions on Signal Processing*, *69*, 2663–2675. https://doi.org/10.1109/TSP.2021.3071210

Xie, S., Yang, Q., Sun, Y., Han, T., Yang, Z., & Shi, Z. (2024, December). Semantic Communication for Efficient Point Cloud Transmission. In *GLOBECOM 2024 - 2024 IEEE Global Communications Conference* (pp. 2948–2953). https://doi.org/10.1109/GLOBECOM52923.2024.10901573

Yang, K., Wang, S., Dai, J., Tan, K., Niu, K., & Zhang, P. (2023, June). WITT: A Wireless Image Transmission Transformer for Semantic Communications. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1–5). https://doi.org/10.1109/ICASSP49357.2023.10094735

Yang, W., Xiong, Z., Yang, Q., Zhang, P., Debbah, M., & Tafazolli, R. (2026). Channel-Adaptive Cross-Modal Generative Semantic Communication for Point Cloud Trans-

mission. *IEEE Transactions on Cognitive Communications and Networking*, *12*, 5983–5998. https://doi.org/10.1109/TCCN.2026.3657061

Ye, J., Chen, Y., Wang, N., & Wang, X. (2022, June). GIFS: Neural Implicit Function for General Shape Representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12819–12829). https://doi.org/10.1109/CVPR52688.2022.01249

Zhang, C., Liu, M., Huang, W., Xu, Y., Xu, Y., & He, D. (2025, April). Deep Joint Source-Channel Coding for Wireless Point Cloud Transmission. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1–5). https://doi.org/10.1109/ICASSP49660.2025.10889738

Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2021, October). Point Transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 16239–16248). https://doi.org/10.1109/ICCV48922.2021.01595

Zhong, X., Pan, Y., Behley, J., & Stachniss, C. (2023, May). SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8371–8377). https://doi.org/10.1109/ICRA48891.2023.10160907