International Journal of Computer Vision
https://doi.org/10.1007/s11263-025-02387-6

Deep Hierarchical Learning for 3D Semantic Segmentation

Chongshou Li[1] · Yuheng Liu[2] · Xinke Li[3] · Yuning Zhang[2] · Tianrui Li[1] · Junsong Yuan[4]

Received: 6 February 2024 / Accepted: 5 February 2025
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

The inherent structure of human cognition facilitates the hierarchical organization of semantic categories for three-dimensional objects, simplifying the visual world into distinct and manageable layers. A vivid example is observed in the animal-taxonomy domain, where distinctions are not only made between broader categories like birds and mammals but also within subcategories such as different bird species, illustrating the depth of human hierarchical processing. This observation bridges to the computational realm as this paper presents deep hierarchical learning (DHL) on 3D data. By formulating a probabilistic representation, our proposed DHL lays a pioneering theoretical foundation for hierarchical learning (HL) in visual tasks. Addressing the primary challenges in effectiveness and generality of DHL for 3D data, we 1) introduce a hierarchical regularization term to connect hierarchical coherence across the predictions with the classification loss; 2) develop a general deep learning framework with a hierarchical embedding fusion module for enhanced hierarchical embedding learning; and 3) devise a novel method for constructing class hierarchies in datasets with non-hierarchical labels, leveraging recent vision language models. A novel hierarchy quality indicator, CH-MOS, supported by questionnaire-based surveys, is developed to evaluate the semantic explainability of the generated class hierarchy for human understanding. Our methodology's validity is confirmed through extensive experiments on multiple datasets for 3D object and scene point cloud semantic segmentation tasks, demonstrating DHL's capability in parsing 3D data across various hierarchical levels. This evidence suggests DHL's potential for broader applicability to a wide range of tasks.

Keywords Hierarchical learning · Point cloud · Semantic segmentation · Deep learning

1 Introduction

Recent advancements in 3D sensing technologies, such as LiDAR and RGB-D cameras, have not only become more accessible and cost-effective but have also spurred significant developments in understanding real-world scenes from 3D data (Bello et al., 2020; Guo et al., 2020a). This progress has been influential in fields ranging from autonomous driving (Cui et al., 2021) to urban planning (Carozza et al., 2014), and from nursing robots (King et al., 2010) to digital twin technologies (Mirzaei et al., 2022) and simulations (Manyoky et al., 2014). At the heart of this advancement is 3D semantic segmentation, essential for classifying fine-grained semantic categories. Despite its progress, driven by recent deep learning advancements, the current paradigm in 3D semantic

Communicated by Bolei Zhou.

✉ Xinke Li
xinkeli@cityu.edu.hk

Chongshou Li
lics@swjtu.edu.cn

Yuheng Liu
sc20yl2@leeds.ac.uk

Yuning Zhang
sc212yz@leeds.ac.uk

Tianrui Li
trli@swjtu.edu.cn

Junsong Yuan
jsyuan@buffalo.edu

[1] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Pidu District, Chengdu 611756, Sichuan, China

[2] SWJTU-Leeds Joint School, Southwest Jiaotong University, Pidu District, Chengdu 611756, Sichuan, China

[3] Department of Data Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

[4] Department of Computer Science and Engineering, University at Buffalo, State University of New York, 12 Capen Hall, Buffalo, NY 14260, USA

Published online: 06 March 2025

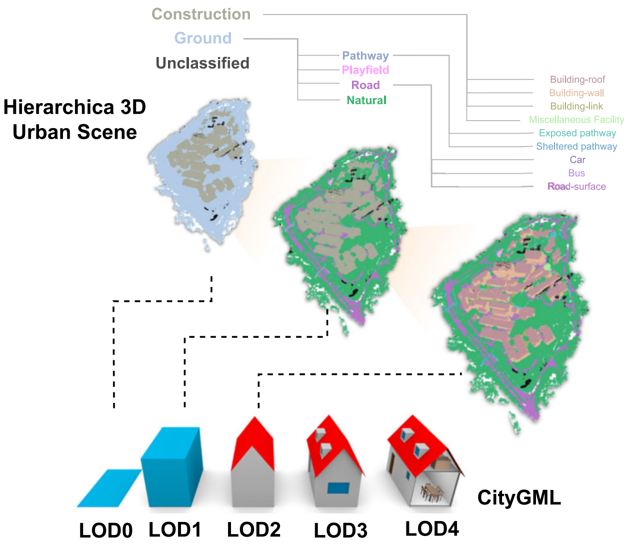


Fig. 1 Examples of class hierarchies in Campus3D (Li et al., 2020). The hierarchal labels from Campus3D are derived from CityGML (Kolbe et al., 2005), which are aligned with the inherent hierarchal nature of real-world modeling. Our 3D DHL is supposed to capture and leverage this structural knowledge by learning

segmentation often struggles with the semantic complexity of real-world data, particularly in handling its hierarchical nature, as depicted in Fig. 1. In contrast, human intelligence excels in interpreting the visual world hierarchically (Uyar et al., 2016; Kaiser et al., 2019). For instance, humans naturally categorize fish and horses under animals, and cars and buses under vehicles. This innate hierarchical inference capability of human intelligence has inspired numerous successful machine learning applications across various domains (Silla & Freitas, 2011).

Conventional 3D semantic segmentation techniques are predominantly based on simple and rigid assumptions: the semantic category of a 3D point is unique and independent, and thus points should be distinctly categorized during prediction (Guo et al., 2020a; Nguyen & Le, 2013). Nevertheless, objects or scenes in the real three-dimensional world often decompose into entities with hierarchical structures, a complexity not encompassed by the aforementioned strategies. For a case in point, "*Ground*" in a 3D scene can be decomposed into "*Pathway*", "*Playfield*", "*Road*" and "*Natural*" (see Fig. 1). Various applications could leverage such hierarchical properties since it allows the same high-resolution 3D data to be used for tasks at different abstraction levels. One practical example is the Level of Detail (LoD), which is defined by the widely-used virtual 3D city format, CityGML, for urban reconstruction (Kolbe et al., 2005). There are five standard levels, represented as LoD0 to LoD4, which enable CityGML to be used in a wide range of applications, from urban planning and noise modeling (which requires lower LoDs) to simulations of Computational Fluid Dynamics (CFD) (which requires moderate LoDs) and then to drone navigation (which requires higher LoDs) (Luebke, 2003). Despite the grave importance of hierarchical semantic structure, this area remains largely unexplored in the existing landscape of the 3D segmentation literature (Li et al., 2022; Guo et al., 2020a; Gao et al., 2021). Only two exceptions stand out: Campus3D (Li et al., 2020) and PartNet (Mo et al., 2019). Their main contributions are to propose the 3D hierarchical semantic segmentation problem by providing hierarchically annotated datasets. Expanding upon the foundations laid by PartNet and Campus3D, this paper presents a novel learning framework called Deep Hierarchical Learning (DHL). The primary aim of DHL is to facilitate a more comprehensive understanding of 3D hierarchy-aware semantic segmentation.

Although the concept of hierarchy-aware 3D semantic segmentation receives limited attention, its associated problem is extensively explored in the realms of machine learning (Li et al., 2022; Silla & Freitas, 2011; Athanasopoulos et al., 2020). These works referred to as Hierarchical Classification which is a multi-label classification problem and classes are hierarchically organized as a tree or a directed acyclic graph (DAG); in these structures, each node corresponds to a semantic label, and edges depict the label dependencies; every data sample is associated with a or multiple root-to-leaf paths within the given class hierarchy (Silla & Freitas, 2011). These efforts are dedicated to ensuring that the predicted labels conform to the DAG relationship in an intuitive manner, the concept of which is referred to as "hierarchical coherence" (Bi & Kwok, 2011). Among them, it is implemented on the loss function either as a regularization term (Wehrmann et al., 2018; Li et al., 2020; Chen et al., 2022) or weighting strategies for penalizing errors individually for classes at different levels of the hierarchy (Bilal et al., 2017; Giunchiglia & Lukasiewicz, 2020; Li et al., 2022; Bertinetto et al., 2020). However, these works relying on ad-hoc design may lack a throughout analysis of the optimality of the solution. Moreover, when applying hierarchical classification to computer vision tasks, especially in fine-grained segmentation tasks, there has been limited exploration beyond the scope of a recent work (Li et al., 2022). It has proposed a straightforward inter-pixel relationship under the context of hierarchical learning but has not scaled from 2D to 3D data. Overall, previous studies have not established a comprehensive and universal analysis of the hierarchical learning problem, and they have substantial limitations in extending the applications to more practical fields such as 3D segmentation.

This paper attempts to outline a general theoretical expression of hierarchical coherence, and thus establish a universal framework for fine-grained 3D vision tasks. Specifically, inspired by hierarchical forecasting (HF) for time series (Athanasopoulos et al., 2023; Hyndman & Athanasopoulos, 2018), we introduce an aggregation matrix of HF to refine a numerical relationship among classes at different hierarchy levels. Utilizing this relationship, we propose the equivalence between the minimization of cross-entropy for classification and the maximization of hierarchical coherence in HL. This equivalence reinterprets the constraints of hierarchical coherence, which were previously intuitive but vague, through the lens of probability. We then abstract a simple yet effective implementation of a hierarchical loss based on it. While the loss function is originally formulated to classify singular samples, *i.e.,* points, a fine-grained task such as semantic segmentation, involving various samples, requires a thorough consideration of the interrelations among multiple points (Nickel & Kiela, 2017).

To address this goal, we design a deep module customized for learning point-wise embeddings that align with cross-points hierarchical relationships. Previous studies primarily focused on constraining sample embeddings by the loss in deep metric learning (Chen et al., 2018; Mousavi et al., 2017; Niu et al., 2017; Yang et al., 2020; Wehrmann et al., 2018). However, applying these techniques to 3D point clouds, which are inherently dense in space and contain numerous samples, is notably resource-intensive and inefficient. Therefore, we propose the novel idea of integrating the information of the hierarchical structure of points directly within the architecture of a deep model. In particular, we propose a hierarchical embedding fusion module (HEFM) module to learn point-wise embeddings that considers two conditions: the top-down coherence condition (TDCC) and the bottom-up coherence condition (BDCC), which are derived from the previous analysis of hierarchical coherence in the context of multiple sample cases. Our subsequent experiments demonstrate that the deep HL method, incorporating the proposed hierarchical coherence loss and HEFM, substantially improves the performance of 3D semantic segmentation across various granularity levels. This advancement is expected to boost applications in 3D scene reconstruction and shape part segmentation.

We also extend our method into a wider dimension beyond the hierarchically annotated tasks. The results of our method in these tasks suggest that incorporating class hierarchy can improve fine-grained segmentation performance, aligning with previous research findings (Chen et al., 2018). This improvement is attributed to the fact that, as mentioned in (Li et al., 2020; Chen et al., 2018), hierarchical structures provide additional guidance for fine-grained classification, addressing geometric ambiguity issues where objects may be geometrically similar but semantically distinct. This leads to a new question: can we generate class hierarchies for datasets that are only annotated with fine-grained labels to aid in task learning? Towards the question, by considering the low scalability and high cost of manually labeling, we propose to generate the semantic hierarchies from the fine-grained classes via leveraging the recent process in the multimodality field. We first use the pretrained vision language model (VLM), such as CLIP (Radford et al., 2021), to derive semantic embeddings of classes that encapsulate both semantic and geometric characteristics. Subsequently, we adapt hierarchical clustering methods to create a class dendrogram, which is further refined into a class hierarchy using large language models (LLM). Due to the comprehensive training datasets of these models, our method exhibits enhanced standardization and universality compared to the potentially biased and nonstandardized views of annotators. In this sense, human review can serve as an evaluative tool to assess the interpretability of the generated class hierarchy rather than participating in its construction. We validate the efficacy of our proposed method through experimental studies on the SensatUrban dataset (Hu et al., 2021), which is characterized by singular annotation.

In summary, we make four major contributions:

- To the best of our knowledge, we are the first to formalize the hierarchical learning problem within a probabilistic representation for 3D vision. This formalization with theoretical results stimulates the establishment of quantitative relationships among different hierarchical levels, providing a foundation for a novel hierarchical coherency loss.
- We introduce a generalized deep framework for conducting 3D semantic segmentation across various hierarchical levels. This framework incorporates a deep module designed to derive point-wise embeddings, capturing hierarchical relationships, and achieving cross-point coherence conditions (*i.e.,* BDCC and TDCC).
- We propose a pragmatic strategy for extracting hierarchical annotations from datasets annotated solely at a fine-grained level. This method leverages recent advancements in multimodal domains (e.g., CLIP) and hierarchical clustering, potentially expanding the applicability of 3D datasets to diverse tasks. Additionally, we have developed a novel class hierarchy quality indicator, class hierarchy mean opinion score (CH-MOS), to evaluates the explainability of the generated class hierarchy for human understanding.
- We validate the effectiveness of our proposed loss function and deep module through point cloud semantic segmentation (PCSS) tasks applied to existing hierarchically-annotated 3D object and scene datasets. Our results demonstrate significant enhancements in PCSS performance, underscoring the utility of our approach. To encourage further development in HL, we will release the source code upon the publication of this paper.

This paper is an extension of the conference *oral presentation* paper (Li et al., 2020). The extension includes the following aspects: *first*, we introduce the new constraints to refine the proposed hierarchical framework in (Li et al., 2020), and prove that minimizing cross-entropy with the constraint is essentially equivalent to maximizing coherence in the hierarchy; *second*, we integrate a novel deep module into the framework which further enhances the point-wise embedding in the hierarchical segmentation task. *finally*, the experimental studies are significantly expanded from the single dataset Campus3D to PartNet (Mo et al., 2019), SensatUrban (Hu et al., 2021) as well as indoor dataset ScanNet200 (Rozenberszki et al., 2022) where piratical solutions of class hierarchy generation for fine-grained annotated datasets (*e.g.,* SensatUrban) is supplied.

The remainder of this paper is structured as follows: Sect. 2 provides a review of related work, followed by the mathematical formulation and essential background knowledge of the problem in Sect. 3. Sect. 4 presents and discusses the framework proposed in this study. Experimental studies and their results are detailed in Sect. 5. Finally, the paper concludes with Sect. 6, summarizing the findings and possible extensions of this work.

2 Related Work

The related work of this paper can be divided into three parts: 1) hierarchical classification, 2) 3D point cloud semantic segmentation and 3) hierarchical forecasting (HF). The details of each part are reviewed as follows.

2.1 Hierarchical Classification

Hierarchical classification is a classical machine learning problem and has been widely applied in different areas (Silla & Freitas, 2011), such as image classification (Deng et al., 2009; Bengio et al., 2010), text classification (Bengio et al., 2010) as well as gene function prediction (Barutcuoglu et al., 2006). In this problem, the class labels are organized in a predefined hierarchy and each data point is associated with one or multiple paths in the hierarchy; the hierarchy can be either a tree or a direct acylic graph (DAG) (Silla & Freitas, 2011; Giunchiglia & Lukasiewicz, 2020; Li et al., 2022). For the problem associated with multiple paths, it is known as hierarchical multilabel classification (HMC) problem. Based on the survey by Silla and Freitas (Silla & Freitas, 2011), approaches can be roughly divided into three groups: flat classification, local classification and global classification. The first method is hierarchy-agnostic and trivial, and it only trains a classifier for the most fine-grained classes (leaf nodes). Then the rest coarse-grained classes are predicted by a bottom-up manner via coherence constraint; the local classification approach first proposed by Koller and Sahami (1997). It trains independent classifiers for each node within taxonomy, and then generate predictions in a top-down manner. This approach results in error propagation problem and various methods have been proposed to solve it (Bennett & Nguyen, 2009; Bi & Kwok, 2015; Ramaswamy et al., 2015; Zhang et al., 2017). Another significant challenge in local classification is determining the positive and negative training examples for each class (node), especially given the need to address class imbalance problem (Eisner et al., 2005; Fagni & Sebastiani, 2007; Xu & Geng, 2019). Unlike localized strategies, the global approach applies a unified classification model for the entirety of a hierarchy's classes. This method has been exemplified in models such as CLUS-HMC (Vens et al., 2008), Clus-Ens (Vens et al., 2008), and in various neural network based methods (Masera & Blanzieri, 2019; Borges & Nievola, 2012). These pioneer models demonstrate the potential of global approaches, offering a holistic perspective towards class hierarchies as opposed to scrutinizing individual class entities, thereby fostering efficient classification procedures.

Within the realm of computer vision, the enhanced capabilities of deep learning fortify the process of hierarchy-aware classification and segmentation. The existing work can be categorized into three primary sections (Li et al., 2022; Bertinetto et al., 2020): 1) hierarchical embedding - which entails both data and label transformation based on the hierarchy (Bengio et al., 2010; Nickel & Kiela, 2017; Chen et al., 2018); 2) hierarchical loss - which focuses on enhancing the consistency across various levels of hierarchy in both training and predictions(Li et al., 2022, 2020; Giunchiglia & Lukasiewicz, 2020), and 3) hierarchical architectures (Mo et al., 2019; Yu et al., 2019; Yan et al., 2015; Zweig & Weinshall, 2007; Wehrmann et al., 2018; Jiang et al., 2019) - which entails the design of neural network layers that draw inspiration from hierarchies.

Our work is the simple path classification and the class taxonomy is a tree. Inspired by preceding efforts, we are the first to explore the inherent link between classification accuracy and coherence. Rather than treating the coherence constraint as a mere regularization item, we delve deeper and establish a theoretical relationship between cross-entropy minimization and coherency maximization. An innovative step in our approach involves the use of an aggregation matrix to quantitatively model and interpret the relationships among different hierarchical levels of classes. This fresh perspective has profound implications for the hierarchy-aware classification.

2.2 Point Cloud Semantic Segmentation

3D point cloud semantic segmentation (PCSS) is a challenging vision task, and it requires multi-granularity features; the methods can be roughly divided into four groups: 1) projection-based methods (Lawin et al., 2017; Audebert et al., 2017; Tatarchenko et al., 2018; Zhang et al., 2020), 2) voxelization-based methods (Choy et al., 2019; Zhu et al., 2021; Rethage et al., 2018), 3) point-based methods (Guo et al., 2020b; Qi et al., 2017b; Choy et al., 2019; Hu et al., 2020; Zhao et al., 2021; Yang et al., 2023; Wu et al., 2024), and 4) hybrid methods (Dai and Nießner, 2018; Liu et al, 2019b; Tang et al, 2020). The first two can leverage the power of deep learning on 2D/organized data via transforming irregular 3D points into regular data. However, they suffer from the problem of information loss. The efficiency of the hybrid methods is low. Our work belongs to the third group, which directly learn unorganized points and is pioneered by PointNet (Qi et al., 2017a). It uses shared multilayer perceptions (MLPs) to extract the per-point features, which is computationally efficient. But it is not able to learn local feature around each point. To address this limitation, various extensions have been proposed and they can be grouped into four categories (Guo et al., 2020a; Hu et al., 2020): 1) neighboring feature pooling (Qi et al., 2017b; Hu et al., 2020; Huang et al., 2018; Zhao et al., 2019), 2) attention based methods (Zhang & Xiao, 2019; Lai et al., 2022), 3) CNN-based methods (Mao et al., 2019; Thomas et al., 2019; Su et al., 2018), and 4) graph-based methods (Liu et al., 2019a; Jiang et al., 2019). Although these studies have achieved good performance, there is no hierarchical relationship among class labels.

Our work is significantly different from the above works. The labels are single-layer while we are processing hierarchical labels which is motivated by LoD in CityGML (Kolbe et al., 2005). Moreover, we have provided a uniform framework for large-scale 3D point cloud segmentation.

2.3 Hierarchical Learning in Visual Tasks

Hierarchical relationships naturally exist in visual understanding tasks, leading to extensive research incorporating such structural information. Therefore, hierarchical methods have since proliferated across various domains: from human parsing (Wang et al., 2019a, 2020) leveraging body-part relationships, to general semantic segmentation using graph neural networks (Li et al., 2022) for structured knowledge propagation. Specifically for HMC of images, modern approaches have explored various techniques including label embedding (Frome et al., 2013; Akata et al., 2015), hierarchy-aware losses (Bertinetto et al., 2020; Zhao et al., 2011), and hierarchical architectures (Yan et al., 2015; Ahmed et al., 2016). Recently, a significant advancement in hierarchical learning came through clustering approaches. While traditional hierarchical clustering methods (Kobren et al., 2017) faced scalability issues, recent gradient-based methods in hyperbolic space (Monath et al., 2019; Chami et al., 2020; Long & van Noord, 2023) have shown superior performance by optimizing continuous relaxations of discrete clustering objectives. These methods leverage hyperbolic geometry's natural ability to represent tree-like structures (Nickel & Kiela, 2017; Sala et al., 2018), enabling both theoretical guarantees and improved empirical results.

However, extending these methods from instance-based to dense prediction tasks remains challenging due to high computational cost of clustering. While initial fsattempts like Deep Hierarchical Semantic Segmentation (Li et al., 2022) propose sampling strategies to manage computational complexity, they face significant limitations for large-scale 3D scenes containing hundreds of thousands of points. The challenge becomes particularly acute when dealing with multiple object instances in 3D point clouds, where existing sampling strategies may fail to capture long-range hierarchical relationships and lack theoretical guarantees for their approximations.

2.4 Hierarchical Forecasting

The HF was first proposed by Orcutt (Orcutt et al., 1968) in studying the information loss in data aggregation. Following it, two main widely studied methods appeared in the literature: a) bottom-up (Shlifer & Wolff, 1979) and b) top-down (Hyndman & Athanasopoulos, 2018); the first generates forecasts of the coarse-grained by summing up that of fine-grained while the second decomposes the coarse grained forecasts to the fined-grained ones. The results are naturally coherent, but they failed to use features of all hierarchical levels. In order to address this limitation, forecast reconciliation has been studied which combines the forecasts to make them coherent (Hyndman et al., 2011; Wickramasuriya et al., 2019; Zhang et al., 2023); both linear and non-linear optimal combinations were reported (Wang et al., 2022; Athanasopoulos et al., 2023). Notably, the hierarchy was represented by a binary matrix which is noted as *aggregation matrix* (Hyndman et al., 2011; Athanasopoulos et al., 2023) which defines how the bottom-level data aggregate to the above level data. The aggregation matrix was adjusted as a constraint matrix (Di Fonzo & Girolimetto, 2022) and binary values were also extended to real values (Athanasopoulos et al., 2020).

Our work is different from the above methods in two aspects: 1) these methods only utilize the hierarchy structure to post-process forecasts instead of integrating the structure into the learning/training process for coherence; 2) the majority of HF works are time series-based regression problems. There is little HF-based research for classification problems, and, to the best of our knowledge, we are the first to apply the binary aggregation matrix to classification tasks (Athanasopoulos et al., 2023).

3 Preliminary

3.1 Class Hierarchy

Let $(\mathcal{Q}, \preccurlyeq)$ denote the class hierarchy, where $\mathcal{Q} = \{c_q\}_{q=1}^{Q}$ and $\preccurlyeq$ represent a finite set of semantic classes and pairwise order relationships between classes, respectively. This hierarchy relationship and its properties are formally defined in the following.

Definition 1 (Super-class/Sub-class) For any $c_p, c_q \in \mathcal{Q}$, $c_p \preccurlyeq c_q$ if c_q is a **super-class** of c_p; alternatively, c_p is a **sub-class** of c_q.

Assumption 1 Given any $1 \leqslant q \leqslant k \leqslant p \leqslant Q$, the order relationship $\preccurlyeq$ satisfies the following three properties.

- **asymmetric**: $\forall q \neq p$, if $c_p \preccurlyeq c_q$ then $c_q \not\preccurlyeq c_p$
- **reflexive**: $c_p \preccurlyeq c_p$
- **transitive**: $\forall p \neq q, p \neq k$ and $k \neq q$, $c_p \preccurlyeq c_k$ and $c_k \preccurlyeq c_q$ imply $c_p \preccurlyeq c_q$

The above definitions and assumptions ensure that set $\mathcal{Q}$ is a partially ordered set. The ordered class set can be structured as a tree by adding a synthetic root node, which is exampled by Fig. 2. Next, we define the concept of a class layer which is a subset of $\mathcal{Q}$.

Definition 2 (Class layer) If a non-singleton set $\Omega \subseteq \mathcal{Q}$ satisfies that any two of classes in Ω cannot be compared by $\preccurlyeq$, then it is a class layer.

In this formulation, we note that multiple class layers can be extracted from class hierarchy $\mathcal{Q}$, and each class set is referred to as a class layer of a hierarchy.

Definition 3 (Super/sub-class layer) Let $\Omega = \{c_p\}_{p=1}^{m}$ and $\Omega' = \{c'_q\}_{q=1}^{m'}$ denote two class layers, if

$$\begin{cases} \forall c'_q \in \Omega', \ \exists! c_p \in \Omega, c'_q \preccurlyeq c_p, \\ \forall c_p \in \Omega, \ \exists c'_q \in \Omega', c'_q \preccurlyeq c_p, \end{cases} \tag{1}$$

then Ω is the **super-class layer** of Ω';alternatively, Ω' is the **sub-class layer** of Ω. We refer to such relationship as $\Omega' \preccurlyeq \Omega$.

Lemma 1 *Super/sub-class layer relationship is transitive.*

In a class hierarchy, we can extract a sequence of class layers that can be ordered by the super-sub-class layer relationship. For instance of the hierarchy in Fig. 2, the extracted layers can be {BD, TR}, {RF, WL, AS, VE, RD} and {RF, WL, AS, CR, BS, MR, PW}. It is noted that distinct layers may share common classes to fulfill the definition of the super/sub layer, such as {RF, WL, AS} in the above examples. Formally, we denote the extracted sequence as $\{\Omega^{(1)}, \cdots, \Omega^{(H)}\}$ with size H, and class layers in the sequence adhering to the following relationship: $\Omega^{(h)} \preccurlyeq \Omega^{(h-1)}$ for $h = 2, \cdots, H$ which is the layer number. We also note that any two class layers in the sequence are super/sub class layers due to Lemma 1.

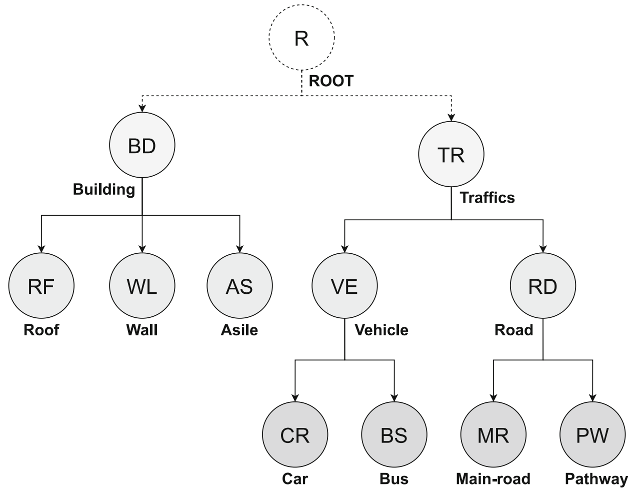


Fig. 2 An example class tree of 11 labels; the class name of each is below each node and, for ease presentation, the short name is inside each node. Each edge represents an order relationship (e.g., RF $\preccurlyeq$ BD, CR $\preccurlyeq$ VE $\preccurlyeq$ TR). An auxiliary root node is added to organize class labels into a tree structure

3.2 Hierarchical Learning

The hierarchical learning (HL) can be further formulated based on the class layers sequence $\{\Omega^{(h)}\}_{h=1}^{H}$. In general, a single-label learning (SL) problem involves learning a predictor parameterized by θ that can output a label distribution $p_\theta(y|\boldsymbol{x})$ for a data sample $\boldsymbol{x}$. The distribution is based on a single class layer. When transitioning to a HL problem, it expands the SL to accommodate multi-predictors scenarios. Namely, HL is to obtain a set of preditors $\{p_{\theta_h}(y|\boldsymbol{x})\}_{h=1}^{H}$ of which each label prediction is based on a class layer in the given sequence, *i.e.,* the predicted labels of $p_{\theta_h}(y|\boldsymbol{x})$ are drawn from $\Omega^{(h)}$. Moreover, the objective of HL is to align the label predictor with ground-truth (GT) label distribution, which for h-th class layer is given by

$$\hat{p}_h(y|\boldsymbol{x}) = \begin{cases} 1 & y = c_{p^{(h)}}, \\ 0 & y \in \Omega^{(h)} \backslash c_{p^{(h)}}, \end{cases} \tag{2}$$

where $c_{p^{(h)}} \in \Omega^{(h)}$ is the annotated label of $\boldsymbol{x}$ for the h-th class layer.

Although the output format of an HL predictor is the same as a combination of multiple single-label predictors across several class layers, the essential difference between HL and a combination lies in the explicit super/sub-class relationship among the GT label distributions. The relationship is specified by

$$\text{if } \prod_{h=1}^{H} \hat{p}_h(c_{q^{(h)}}|\boldsymbol{x}) = 1, \tag{3}$$

$$\text{then } c_{q^{(h)}} \preccurlyeq c_{q^{(h-1)}}, \forall\, h = 2, \cdots, H, \tag{4}$$

which implies the set of predicted labels drawn from GT label distributions is indeed a path of the tree showcased by Fig. 2. Therefore, effective hierarchical learning necessitates incorporating this relationship into the learning process, the crux of which lies in the following concept of hierarchical coherence for predictions.

Definition 4 (Coherence Score) Given two predictors $p_{\theta'}$ and p_θ on two class layers, the coherence score $\kappa_{\theta,\theta'}$ of one sample $\boldsymbol{x}$ is defined as

$$\kappa_{\theta,\theta'}(\boldsymbol{x}) = \mathbb{E}_{\substack{c\sim p_\theta(y|\boldsymbol{x}) \\ c'\sim p_{\theta'}(y|\boldsymbol{x})}} [\mathbb{1}(c \preccurlyeq c')], \tag{5}$$

where $\mathbb{1}(\cdot)$ is the indicator function. If $\kappa_{\theta,\theta'}(\boldsymbol{x}) = 1$, we say $p_{\theta'}$ and p_θ are **hierarchically coherent (HC) predictions** for $\boldsymbol{x}$.

The aforementioned definitions elucidate the predictions of two distinct layers within the HL framework. If these predictions are HC, then predicted class labels drawn from these distributions would satisfy the super/sub class relationship. Furthermore, it is easy to conclude the following lemma of the GT label distributions of an arbitrary sample.

Lemma 2 *Given the GT distributions of a HL problem denoted by* $\hat{p}_1, \cdots, \hat{p}_h, \cdots, \hat{p}_H$*,* $\hat{p}_{h-1}$ *and* $\hat{p}_h$ *are HC predictions for any samples for* $2 \leqslant h \leqslant H$*.*

The Lemma can be easily proved by substituting (2) and (3) to definition (5). Using this Lemma, we can incorporate a constraint, *i.e.,* , the HC constraint, into the HL problem, which aids the learning process in understanding the GT distributions. Subsequently, we demonstrate the following properties of HC predictions which are necessary conditions of HC.

Lemma 3 *Given two class layers* $\Omega = \{c_p\}_{p=1}^{m}$ *and* $\Omega' = \{c'_q\}_{q=1}^{m'}$*, without loss of generality, we assume* $\Omega' \preccurlyeq \Omega$*, if the two predictors* p_θ *and* $p_{\theta'}$ *based on* Ω *and* Ω' *separately are HC predictions for a sample* $\boldsymbol{x}$*, the following bottom-dominated coherence constraint (BDCC) and top-dominated coherence constraint (TDCC) hold:*

- ***BDCC****: if the predicted probability of a class* c'_q *in* Ω' *is 1, then the probability of the class in* Ω *being super-class of* c'_q *is 1,* i.e.,

$$p_{\theta'}(c'_q|\boldsymbol{x}) = 1 \Rightarrow p_\theta(c_p|\boldsymbol{x}) = 1, \forall c'_q \preccurlyeq c_p. \tag{6}$$

- ***TDCC****: if the predicted probability of a class* c_p *in* Ω *is 0, then the probability of any classes in* Ω' *being sub-class of* c_p *is 0,* i.e.,

$$p_\theta(c_p|\boldsymbol{x}) = 0 \Rightarrow p_{\theta'}(c'_q|\boldsymbol{x}) = 0, \forall c'_q \preccurlyeq c_p. \tag{7}$$

The proof is referred to Appendix A. This Lemma introduces two more detailed properties of HC predictions. Considering a sequence of class layers in the HL problem, two predictors associated with $\Omega^{(h-1)}$ and $\Omega^{(h)}$ are expected to be trained to fulfill the above constraints.

4 Methodology

In this section, we present our solution to the HL problem, with a particular focus on addressing HC as discussed previously. To address these challenges, we have developed a novel deep framework, illustrated in Fig. 3. The framework contains two key parts for HL: (1) a loss function aiming at the HC constraint; (2) a deep architecture focusing on the cross-point hierarchical relationship constraint.

4.1 Overall Architecture

The proposed framework depicted in Fig. 3 takes the raw point cloud as input and facilitates the learning of hierarchically coherent predictions. It consists of three key components:

- **Multi-task Network (MTN)**: We employs a shared point-wise encoder-decoder coupled with multiple task-specific prediction heads, where each head corresponds to a distinct hierarchical label level. The MTN architecture enables simultaneous prediction across multiple class layers through dedicated task heads while leveraging shared geometric features from the encoder.
- **Hierarchical Coherence (HC) Loss**: Our proposed network is trained end-to-end by minimizing a combination of the HC loss and cross-entropy loss, where the HC loss is derived from the Theorem.
- **Hierarchical Embedding Fusion Module (HEFM)**: The HEFM leverages structured knowledge (*i.e.,* TDCC and BDCC) to refine point embeddings, generating hierarchy-aware point representations and enhancing

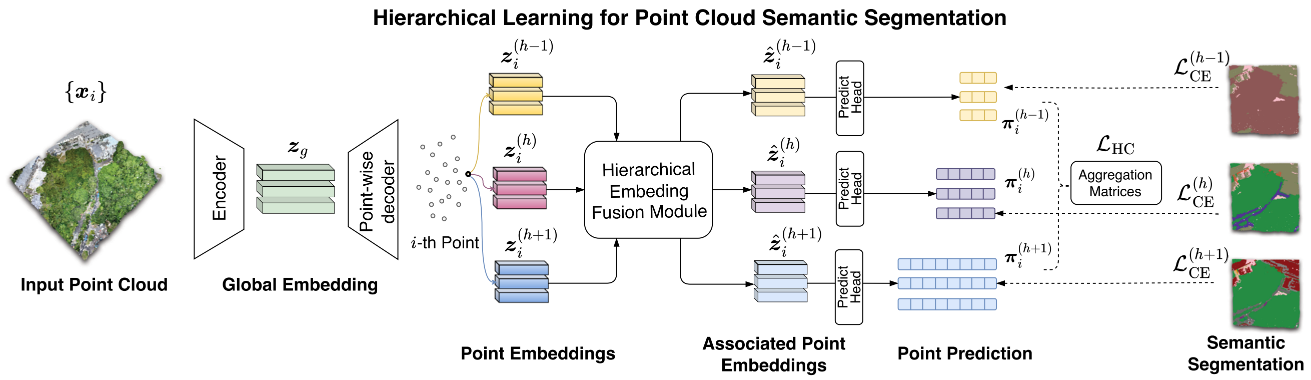

Hierarchical Learning for Point Cloud Semantic Segmentation
$\{\boldsymbol{x}_i\}$
Input Point Cloud
Encoder
$\boldsymbol{z}_g$
Global Embedding
Point-wise decoder
i-th Point
$\boldsymbol{z}_i^{(h-1)}$
$\boldsymbol{z}_i^{(h)}$
$\boldsymbol{z}_i^{(h+1)}$
Point Embeddings
Hierarchical Embedding Fusion Module
$\hat{\boldsymbol{z}}_i^{(h-1)}$
$\hat{\boldsymbol{z}}_i^{(h)}$
$\hat{\boldsymbol{z}}_i^{(h+1)}$
Predict Head
Predict Head
Predict Head
Associated Point Embeddings
$\boldsymbol{\pi}_i^{(h-1)}$
$\boldsymbol{\pi}_i^{(h)}$
$\boldsymbol{\pi}_i^{(h+1)}$
Point Prediction
$\mathcal{L}_{\text{HC}}$
Aggregation Matrices
$\mathcal{L}_{\text{CE}}^{(h-1)}$
$\mathcal{L}_{\text{CE}}^{(h)}$
$\mathcal{L}_{\text{CE}}^{(h+1)}$
Semantic Segmentation

Fig. 3 Architect of HL. It consists of three parts: Multi-task Network (MTN), hierarchical embedding fusion module (HEFM) and hierarchical coherence (HC) loss

coherence in predictions. The architecture is displayed in Fig. 4.

4.2 Hierarchical Coherence Loss

Drawing inspiration from hierarchical regression (Athanasopoulos et al., 2023), we propose the use of an aggregation matrix (AM) to establish a quantitative relationship among predictions at different hierarchical layers. To construct the HC loss function, we first formally define the AM which builds the quantitative relationship among difference class hierarchy layers.

Definition 5 (Aggregation Matrix) Given a class set $\Omega = \{c_p\}_{p=1}^{m}$ is the super-class set of class set $\Omega' = \{c'_q\}_{q=1}^{m'}$, the aggregation matrix is $\mathbf{A}_{\Omega^{(\mathrm{h-1})},\Omega^{(\mathrm{h})}} = [a_{p,q}]_{m\times m'} \in \{1,0\}^{m\times m'}$ associated with Ω and Ω', of which the element is given by

$$a_{p,q} = \mathbb{1}(c'_q \preccurlyeq c_p). \tag{8}$$

Take the class hierarchy of Fig. 2 as an example. If $\Omega' = \{\text{RF, WL, AS, VE, RD}\}$ and $\Omega = \{\text{BD, TR}\}$, the corresponding AM is:

$$\mathbf{A}_{\Omega,\Omega'} = \begin{array}{c} \\ \text{BD} \\ \text{TR} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{RF} & \text{WL} & \text{AS} & \text{VE} & \text{BD} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

Based on the definition of AM, we present the following theorem.

Theorem *Given two predictions of for a sample $\boldsymbol{x}$ $\boldsymbol{\pi}_\theta(\boldsymbol{x})$ and $\boldsymbol{\pi}_{\theta'}(\boldsymbol{x})$ defined on layers $\Omega = \{c_i\}_{i=1}^{m}$ and $\Omega' = \{c'_j\}_{j=1}^{m'}$, they are denoted by*

$$\boldsymbol{\pi}_\theta(\boldsymbol{x}) = [p_\theta(c_1|\boldsymbol{x}), \cdots, p_\theta(c_m|\boldsymbol{x})]^\top,$$
$$\boldsymbol{\pi}_{\theta'}(\boldsymbol{x}) = [p_{\theta'}(c'_1|\boldsymbol{x}), \cdots, p_{\theta'}(c'_{m'}|\boldsymbol{x})]^\top.$$

Suppose the aggregation matrix between Ω and Ω' denoted by $\mathbf{A}$, if the predictions satisfy that

$$|\boldsymbol{\pi}_\theta(\boldsymbol{x}) - \mathbf{A}\boldsymbol{\pi}_{\theta'}(\boldsymbol{x})| = 0, \tag{9}$$

then the entropy $H_\theta(y|\boldsymbol{x}) \to 0$ implies $\kappa_{\theta,\theta'}(\boldsymbol{x}) \to 1$, where $H_\theta(y|\boldsymbol{x}) = \boldsymbol{\pi}_\theta(\boldsymbol{x})^\top \log \boldsymbol{\pi}_\theta(\boldsymbol{x})$.

The proof is provided in Appendix B. In essence, the theorem postulates a crucial conclusion within HL. It states that when a prediction $\boldsymbol{\pi}_\theta$ based on the super-class layer is deterministic (*e.g.,* the GT prediction) and its corresponding sub-class layer prediction $\boldsymbol{\pi}_{\theta'}$, satisfying equation (9), this pair of predictions is HC. Based on this theorem, the HL loss is divided into two parts. The first part is the classification loss, referring to the cross entropy (CE) loss across multiple class layers for each point, which is given by

$$\mathcal{L}_{\text{CE}}(\boldsymbol{x}) = -\sum_{h=1}^{H}\sum_{y\in\Omega^{(h)}} \hat{p}_h(y|\boldsymbol{x}) \log p_{\theta_h}(y|\boldsymbol{x}). \tag{10}$$

where $\hat{p}_h$ and p_{θ_h} are the predicted distribution for data $\boldsymbol{x}$ in the h-th layer. The second part of the loss is a regularization loss, inspired by (9), known as the hierarchical coherence (HC) loss, represented as

$$\mathcal{L}_{\text{HC}}(\boldsymbol{x}) = \sum_{h=2}^{H} \left\| \boldsymbol{\pi}_{\theta_h}(\boldsymbol{x}) - \mathbf{A}_{\Omega^{(h-1)},\Omega^{(h)}} \boldsymbol{\pi}_{\theta_{h-1}}(\boldsymbol{x}) \right\|^2, \tag{11}$$

where $\mathbf{A}_{\Omega^{(h-1)},\Omega^{(h)}}$ is the aggregation matrix between the h and $h-1$ class layers in HL, the vector $\boldsymbol{\pi}_{\theta_h}(\boldsymbol{x}) = [p_{\theta_h}(c_1|\boldsymbol{x}), \cdots, p_{\theta_h}(c_m|\boldsymbol{x})]^\top$ represents the prediction in

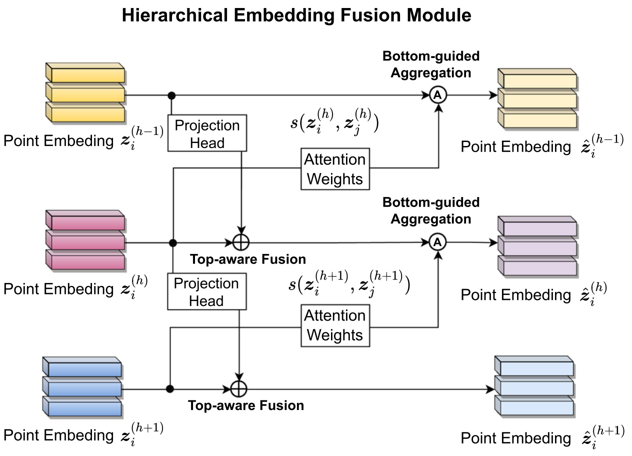

Hierarchical Embedding Fusion Module
Bottom-guided Aggregation
A
Point Embeding $z_i^{(h-1)}$
Projection Head
$s(z_i^{(h)}, z_j^{(h)})$
Attention Weights
Point Embeding $\hat{z}_i^{(h-1)}$
Bottom-guided Aggregation
A
Top-aware Fusion
Point Embeding $z_i^{(h)}$
Projection Head
$s(z_i^{(h+1)}, z_j^{(h+1)})$
Attention Weights
Point Embeding $\hat{z}_i^{(h)}$
Top-aware Fusion
Point Embeding $z_i^{(h+1)}$
Point Embeding $\hat{z}_i^{(h+1)}$

Fig. 4 Architecture of HEFM. There are two components: 1) top-aware fusion and 2) bottom-guided aggregation

layer h, and similarly $\pi_{\theta_{h-1}}$ is for layer $h-1$. Finally, the loss at a single point $\boldsymbol{x}$ is

$$\mathcal{L}(\boldsymbol{x}) = \mathcal{L}_{\text{CE}}(\boldsymbol{x}) + \lambda \mathcal{L}_{\text{HC}}(\boldsymbol{x}), \tag{12}$$

where λ is a balancing parameter. The total loss is the expectation of $\mathcal{L}(\boldsymbol{x})$ for all $\boldsymbol{x}$ in the point clouds.

4.3 Hierarchical Embedding Fusion

The TDCC and BDCC constraints, as defined in Lemma 3, outline the necessary conditions for hierarchically coherent predictions. We contemplate employing them as constraints for HL. However, direct utilization of TDCC and BDCC is redundant regarding the loss in Sect. 4.2. We consider adapting them into soft constraints in terms of multiple samples, *i.e.,* inter-points constraints.

Proposition 1 *Given the condition in Lemma 3 and two samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, following BDCC and TDCC hold:*

- ***BDCC:*** *if $p_{\theta'}(c'_q|\boldsymbol{x}_1)p_{\theta'}(c'_q|\boldsymbol{x}_2) = 1$, then*

$$p_\theta(c_p|\boldsymbol{x}_1)p_\theta(c_p|\boldsymbol{x}_2) = 1, \forall c'_q \preccurlyeq c_p. \tag{13}$$

- ***TDCC:*** *if $p_\theta(c_p|\boldsymbol{x}_1)p_\theta(c_p|\boldsymbol{x}_2) = 0$, then*

$$p_{\theta'}(c'_q|\boldsymbol{x}_1)p_{\theta'}(c'_q|\boldsymbol{x}_2) = 0, \forall c'_q \preccurlyeq c_p. \tag{14}$$

The specific expression of this adaptation is that if two points have inconsistent predictions at the super-class layer, their predictions at the sub-class layer must be inconsistent as well; if their predictions at the sub-class layer are consistent, then their predictions at the super-class layer must also be consistent. These constraints are integrated into the Hierarchical Embedding Fusion Module (HEFM), illustrated in Fig. 4. The HEFM comprises two essential components:

- **Top-aware Fusion**: It ensures that points associated with the same super-class labels in the top-level should be proximate to each other within the embedding space of the bottom-level.
- **Bottom-guided Aggregation**: It enforces that points linked to the same subclass labels in the bottom-level should exhibit similarity within the top-level embedding space.

The core idea behind HEFM is to regulate point embeddings in a way that aligns with these constraints, rather than relying solely on pointwise constraints as specified in Lemma 3 which is computationally expensive. Now we introduce the implementation detail. Note that the HEFM takes the decoder's output as the input. Specifically, we denote the point embedding of the i-th point by the top-level decoder as $z_i^{(h-1)}$, while the point embedding by the bottom-level decoder is denoted as $z_i^{(h)}$.

Top-aware Fusion: New embedding of the bottom-level is obtained by:

$$\hat{z}_i^{(h)} = \alpha z_i^{(h)} + (1-\alpha)\text{Proj}\left(z_i^{(h-1)}\right) \tag{15}$$

where Proj$(\cdot)$ is a projection head which is a simple MLP with batch normalization, and $\alpha \in [0, 1]$ is a tunable factor. In this soft and learnable way, points belonging to different parent categories are further repelled in fine-grained feature space, whereas same parent category points are rarely affected. The $\hat{z}_i^{(h)}$ is used to generate final bottom-level embeddings.

Bottom-guided Aggregation: New embedding of top-level is obtained by:

$$\hat{z}_i^{(h-1)} = \sum_{j=1}^{N} \phi\left(z_i^{(h)}, z_j^{(h)}\right) z_i^{(h-1)}, \tag{16}$$

where $s(\cdot)$ is an attention score function which generates a score based on the similarity between inputs. We use soft attention in our implementation. Considering the large amount of points in a scene, we propose to convert the aggregation into a local version which reduces a considerable computation overhead, which is denoted by

$$\hat{z}_i^{(h-1)} = \sum_{j \in \mathcal{N}_i} \phi\left(z_i^{(h)}, z_j^{(h)}\right) z_i^{(h-1)} \tag{17}$$

where $\mathcal{N}_i$ is the neighborhood point indices of the i-th point and $s(\cdot)$ is in the form of a local attention score function, given by

$$\phi\left(z_i^{(h)},\ z_j^{(h)}\right) = \frac{\exp\left({z_i^{(h)}}^{\top} z_j^{(h)}/\tau\right)}{\sum_{j'\in\mathcal{N}_i}\exp\left({z_i^{(h)}}^{\top} z_{j'}^{(h)}/\tau\right)} \tag{18}$$

with a set parameter τ. In practice, $s(\cdot)$ measures the similarity of the underlying embeddings, reintegrating those top-level embeddings by these similarities. Therefore, the similarity of two points in the bottom-level embeddings will generate highly correlated top-level embeddings, which subtly implements the constraint of TDCC. In our implementation, the neighbor size $|\mathcal{N}_i|$ is set to 40. An analysis of computational cost is presented in Sect. 5.6.

4.4 Class Hierarchy Mining

The preceding sections detailed a method for training segmentation models on hierarchically annotated 3D datasets. Yet, the significant annotation effort required for such datasets often limits their availability (Li et al., 2020; Mo et al., 2019). To broaden our approach's applicability, we introduce a technique to construct a class hierarchy from the fine-grained class labels within the dataset. Our methodology exploits a Vision Language Model (VLM) to derive class embeddings. Subsequently, these embeddings inform a hierarchical clustering process, producing a dendrogram of classes. It's essential to underscore that constructing a label taxonomy for hierarchical 3D segmentation should account for both the semantic relevance and the geometric characteristics of the class. Recent advances in VLMs, particularly in aligning geometric features in images with language embeddings, enable us to harness pre-trained embeddings for this hierarchy extraction. We utilize the widely-acknowledged CLIP text encoder (Radford et al., 2021), train on the WebImage Text Dataset, to encode the fine-grained classes present in the 3D dataset. For improved precision, we employ the dataset's class definitions as caption text, rather than solely the class terms, thus mitigating potential word embedding ambiguities. These embeddings then serve as features, guiding the iterative merging of clusters from the original fine-grained classes into a class dendrogram. Ultimately, we employ a large language model (LLM) to prune the generated dendrogram, resulting in a semantically coherent class hierarchy. This entire process is depicted in Fig. 5.

Let us delve deeper into the hierarchical agglomerative clustering method we applied. Denote the set of fine-grained classes as $\Omega^{(H)}$. For each class $c_i \in \Omega^{(H)}$, its corresponding embedding produced by the VLM is given by $\boldsymbol{\mu}_i \in \mathbb{R}^d$, where d signifies the dimensionality of the embedding space. The foundational idea of our clustering approach is to continually merge the two closest clusters based on certain distance metrics until only a single cluster remains. At the outset, every fine-grained class label constitutes its own singleton cluster. For any two clusters, $\{c_i\}$ and $\{c_j\}$ (where $c_i, c_j \in \Omega^{(H)}$), their inter-distance d_{ij} is characterized using the cosine similarity:

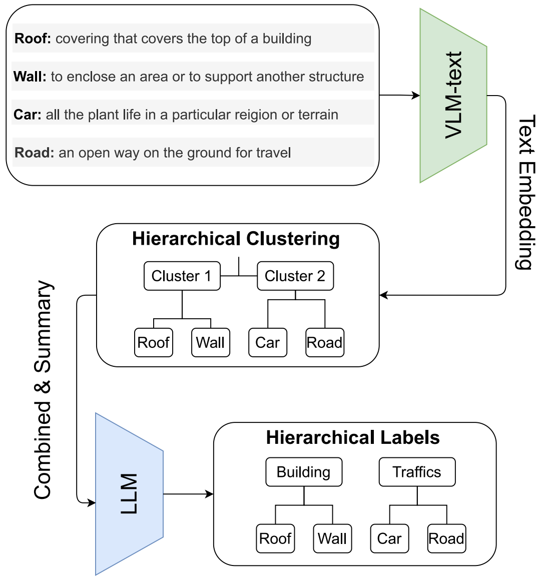

Roof: covering that covers the top of a building
Wall: to enclose an area or to support another structure
Car: all the plant life in a particular region or terrain
Road: an open way on the ground for travel
VLM-text
Text Embedding
Hierarchical Clustering
Cluster 1
Cluster 2
Roof
Wall
Car
Road
Combined & Summary
LLM
Hierarchical Labels
Building
Traffics
Roof
Wall
Car
Road

Fig. 5 Overflow of Label Hierarchy Mining from Fine-grained Classes

$$d_{ij} = \frac{\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_i\|\|\boldsymbol{\mu}_j\|}. \tag{19}$$

During each iteration, only the two closest clusters are merged. Subsequently, the distance between this newly formed cluster and the existing ones is calculated using Ward's method (Ward Jr, 1963). For instance, if in the first iteration, clusters $\{c_i\}$ and $\{c_j\}$ are merged to form a new cluster $\{c_i, c_j\}$, the distance between this new cluster and another singleton cluster $\{c_k\}$ (where $k \neq i$ and $k \neq j$) is:

$$\begin{aligned} d_{i*k} = & \frac{s_i + s_k}{s_i + s_j + s_k} d_{ik} + \frac{s_j + s_k}{s_i + s_j + s_k} d_{kj} \\ & - \frac{s_i + s_j}{s_i + s_j + s_k} d_{ij}, \end{aligned} \tag{20}$$

here $i*$ denotes the index of the newly formed cluster $\{c_i, c_j\}$, while s_i, s_j, and s_k represent the sizes of clusters $\{c_i\}$, $\{c_j\}$, and $\{c_k\}$ respectively. A comprehensive breakdown of the clustering method is offered in Algorithm 1. Lastly, it's important to note that the total number of iterations is equal to $|\Omega^{(H)}| - 1$, given that in each step, only two clusters merge until a solitary cluster remains.

Algorithm 1 will generate a dendrogram with the height of $|\Omega^{(H)}| - 1$. We apply an empirical strategy to truncate the dendrogram for class clusters extraction, where the truncation threshold is set a hierarchy height of $\frac{|\Omega^{(H)}|-1}{2}$. To ensure

Algorithm 1 Fine-grained Class Clustering Algorithm

```
procedure CLUSTERING ($\Omega^{(H)}, \{\boldsymbol{\mu}_i | c_i \in \Omega^{(H)}\}$)
    Initialize number of iterations: $h \leftarrow |\Omega^{(H)}| - 1$
    Construct a set of clusters: $\Phi^{(h)} = \{\{c_i\} | c_i \in \Omega^{(H)}\}$
    Initialize cluster pairwise distance $d_{ij}$ by (19)
    while $h > 0$ do
        Merge closest clusters $i, j$ at $\Phi^{(h)}$ as cluster $i^*$
        Update cluster set:
            $\Phi^{(h-1)} \leftarrow \Phi^{(h)} \cup \{\{c_i, c_j\}\} \setminus \{\{c_i\}, \{c_j\}\}$
        Compute the distance $d_{i^*k}$ by (20)
        $h \leftarrow h - 1$
    end while
    Return clustering result: $\{\Phi^{(h)} | h = 0, \ldots, |\Omega^{(H)}| - 1\}$
end procedure
```

the class cluster extracted from dendrogram is semantically explainable, we utilize the large language model (LLM), specifically GPT (Brown et al., 2020) to generate the semantic label of each cluster. We imply exclusive rules in cluster labels to avoid ambiguities. In the prompt, we add the following instruction: "......*ensuring the names are mutually exclusive in semantics*". The specific prompt used to construct cluster names for the class hierarchy of SensatUrban dataset (Hu et al., 2021) is provided in the Appendix.

5 Experimental Studies

We present a comprehensive framework designed to manage the hierarchical PCSS. This framework, grounded in MTN architecture, incorporates HC loss for hierarchical segmentation and HEFM for point-wise embedding learning. To appraise the effectiveness of our proposed framework, we administer a series of rigorous experiments on semantic segmentation within existing 3D scene-based datasets with hierarchical annotations, including Campus3D (Li et al., 2020) and ScanNet200 (Brown et al., 2020), as well as hierarchical part segmentation on object-based PartNet (Mo et al., 2019). Our results confirm that our methods enhance the performance metrics of PCSS. Furthermore, to address the scarcity of hierarchical datasets in 3D, we apply the proposed hierarchy mining algorithm that autonomously generates hierarchical annotations for datasets with single fine-grained label annotations, namely, SensatUrban (Hu et al., 2021). This generation methodology capitalizes on the recent advancements in vision-language integration and large language models. To demonstrate its efficacy, we perform the algorithm and execute a comprehensive analysis on single-layer annotated point cloud datasets.

5.1 Experimental Setting

Network Architecture. In our experiments, we employ the following architectures for point cloud segmentation: SparseUNet (Graham et al., 2018), RandLANet (Hu et al., 2020), PointNet++ (Qi et al., 2017b), MinkowskiUNet (Choy et al., 2019), Point Transformer (PT) (Zhao et al., 2021), Point Transformer V3 (PTv3) (Wu et al., 2024), and Swin3D (Yang et al., 2023). Following the proposed paradigm in the original works, Swin3D and PT are only implemented for indoor scene segmentation, while RandLANet is only implemented for outdoor scene. The implementation of all architectures follow either the released code base (Contributors, 2023) or their official implementations.

Hardware Configurations. We note that all experiments are run on NVIDIA GeForce RTX 3090 GPUs and Linux OS with Ubuntu 21.04 version, Intel(R) Xeon(R) Gold 6330 CPU.

Training Settings. For training on outdoor scene-based datasets, specifically Campus3D and SensatUrban, we adopt standard pipeline with hyperparameters from (Li et al., 2020). For training on indoor scene-based datasets and object-based datasets, namely, ScanNet200 and PartNet, we follow the settings including optimizer, scheduler, and data loader in (Contributors, 2023) (PartNet follows the setting of the common ShapeNet-Part dataset (Yi et al., 2016)), except for the batch size adapted for our hardware memory.

Implementation Details. As specified in Sect. 4, we implement an Multi-layer Perception (MLP) multi-head configuration for MTN across each architecture. However, we set an exception in SparseUNet (Graham et al., 2018) where employing a Multi-decoder setup-leveraging multiple decoder structures within UNet-yields superior performance compared to MLP multi-heads, as demonstrated in Table 7. We attribute this to the smaller dimensionality of point embeddings in SparseUNet relative to point-based networks, resulting in suboptimal MLP performance compared to Multi-decoders. To maintain network generality and avoid ad-hoc modifications, we adjusted only this network in the following experiments while keeping the multi-head approach for the others. Additionally, we pre-generate the nearest indices for each point in point clouds before the training to bypass the computational cost of neighbor search induced by (16) in HFEM.

5.2 Evaluation Protocol

In line with established PCSS studies (Xie et al., 2020; Guo et al., 2020b), we utilize the Overall Accuracy (OA) and mean Intersection Over Union (mIoU) across all classes as our primary evaluation metrics. For each individual class, IoU is determined using the formula $\frac{\text{TP}}{\text{T+TP−P}}$, where TP stands for the number of true positive points, T represents the total ground truth points attributed to the respective class, and P signifies the count of predicted positive points. Notably, we evaluate OAs and mIoUs across multiple class layers present in hierarchically annotated datasets. For a more detailed comparative analysis, we consider metrics derived from training a network separately at each hierarchical level as a benchmark, and we refer to this as the "multi-classifier" approach. The disparity in performance between the multi-classifier method and the HL offers insights into the impact of the HL. Additionally, to assess hierarchical coherence, we calculate the coherence score as per Definition 4, and we extend the original formulation of two layers to the case of H layers:

$$\kappa_{\theta_1,\ldots,\theta_H}(\boldsymbol{x}) = \mathbb{E}_{c^{(1)}\sim p_{\theta_1}(y|\boldsymbol{x}),\ldots,c^{(H)}\sim p_{\theta_H}(y|\boldsymbol{x})}[\mathbb{1}(c^{(1)} \preccurlyeq \ldots c^{(H)})],$$

where $c^{(1)}, \cdots, c^{(H)}$ are classes in layers $\Omega^{(1)}, \cdots, \Omega^{(H)}$, respectively. Furthermore, we average the scores across all samples and present the average value as the final coherence score. The efficacy of the HL in 3D segmentation is further elucidated by a combined analysis of these three performance indicators across three distinct datasets.

To evaluate the quality of mined class hierarchy quality, we propose an indicator, **class hierarchy mean opinion score (CH-MOS)**, which is based on the widely used mean opinion score (MOS) in the area of telecommunications engineering and Quality of Experience (Viswanathan & Viswanathan, 2005). The CH-MOS indicator consists of five dimensions: **intra-class similarity (IaCS)**, **inter-class separation (IeCS)**, **applicability (AP)**, and **semantic validity of cluster names (SVCN)**; and each dimension is rated by the MOS with a scale of 1 to 5. The CH-MOS is the average score over these four dimensions. Detailed information on these metrics is referred to in Section 3 of the Appendix. Evaluation results are given at Sect. 5.4.1.

5.3 Results

5.3.1 Semantic Segmentation on Campus3D

The Campus3D dataset (Li et al., 2020) is the first photogrammetry point cloud dataset specifically designed for deep learning-based hierarchical segmentation. It contains 6 outdoor scenes, each consisting of nearly 100 million points, with each point being hierarchically annotated with semantic labels. This hierarchical annotation is ideally suited for our proposed HL method. To alleviate the computational overhead of processing the large scene and heavy label imbalance, we reprocessed the Campus3D dataset using the method from the SensatUrban (Hu et al., 2021). We divided it into 26 square areas, each measuring $200m \times 200m$, and allocated these areas into training, validation, and testing sets at a ratio

Table 1 Hierarchical semantic segmentation results on Campus3D (Li et al., 2020) dataset. "**(+)**" and "*(−)*" stand for the positive and negative gain of metrics by HL method

Model	HL	mIoU^1	mIoU^2	mIoU^3	OA^1	OA^2	OA^3
PointNet++ (Qi et al., 2017b)	w/o	92.9	42.5	41.0	95.8	89.6	81.6
	w/	92.0 *(−0.9)*	55.0 **(+12.6)**	41.2 **(+0.3)**	94.5 *(−1.3)*	93.1 **(+3.5)**	82.9 **(+1.3)**
RandLANet (Hu et al., 2020)	w/o	86.0	51.3	27.4	91.1	89.9	76.0
	w/	92.1 **(+6.1)**	61.2 **(+9.9)**	41.8 **(+14.5)**	96.4 **(+5.2)**	91.4 **(+1.5)**	90.8 **(+14.8)**
SparseUNet (Graham et al., 2018)	w/o	86.0	55.2	39.4	90.3	88.3	85.0
	w/	87.8 **(+1.8)**	57.2 **(+2.0)**	40.7 **(+1.3)**	91.0 **(+0.7)**	89.0 **(+0.6)**	85.6 **(+0.6)**
MinkovskiUNet (Choy et al., 2019)	w/o	88.9	46.7	36.1	91.5	86.5	80.3
	w/	91.1 **(+2.2)**	55.5**(+7.8)**	38.8 **(+2.7)**	93.4**(+1.9)**	91.4 **(+4.9)**	84.7 **(+4.4)**
PointTransformerV3 (Wu et al., 2024)	w/o	92.9	57.8	44.1	94.5	90.0	85.2
	w/	93.8**(+0.9)**	61.3 **(+3.5)**	45.2 **(+1.1)**	95.4 **(+0.9)**	92.5 **(+2.5)**	86.3 **(+1.1)**

Bold values indicate improvements by HL

The numerals 1, 2, and 3 correspond to the three hierarchical levels of segmentation granularity, denoting coarse-, middle-, and fine-grained categories, respectively. This notation is consistent throughout all subsequent mentions

Table 2 Average mIoU (mIoU-A) and OA (OA-A) across three levels of Campus3D (Li et al., 2020)

Model	mIoU-A	OA-A
DGCNN (Wang et al., 2019b)	58.9	87.9
PCNN (Li et al., 2018)	58.1	86.4
PointNet++ (Qi et al., 2017b)	58.8	88.3
SparseUNet (Graham et al., 2018)	60.2	88.0
RandLANet (Hu et al., 2020)	54.8	85.7
PointTransformerV3 (Wu et al., 2024)	64.8	89.8
PointTransformerV3+HL (Ours)	**66.7**	**91.4**

Bold values indicate the best performance

of 20/3/3. Additionally, we merge the three least prevalent labels in Campus3D into a "miscellaneous facility" category. In the end, we apply 3 hierarchical levels applied in the experiments. The detailed hierarchy is referred to in the Appendix.

The backbone models used here are: PointNet++ (Qi et al., 2017b), RandLANet (Hu et al., 2020), SparseUNet (Graham et al., 2018), MinkowskiUNet (Choy et al., 2019), and Point Transformer V3 (PTv3) (Wu et al., 2024). The results of mIoU and OA for these five backbone models on the Campus3D dataset are presented in Table 1. Across nearly all hierarchical levels and models, we observe consistent performance gains when employing HL. The sole deviation from this trend is with the multi-classifier variant of PointNet++ (i.e., PointNet++ without HL), which outperforms its HL counterpart at the most coarse-grained level. This can be attributed to the fact that PointNet++ is a network with a simple structure, making it less suited for complex tasks and limiting its representational capabilities. In HL, when the three tasks are combined, the more challenging fine-grained learning task can impede the learning of the simpler coarse-grained task, resulting in limited enhancement of PointNet++'s performance. However, the HL does provide a significantly stronger boost for more complex networks like PTv3. This improvement is likely due to the intrinsic relationships among hierarchical label layers, which may offer supplementary geometric information beneficial for semantic segmentation. This is visually demonstrated in Fig. 6, where HL-equipped models effectively resolve cases of *geometric ambiguity*-situations where geometrically similar structures are semantically distinct, as discussed in (Li et al., 2020). In contrast, models without HL struggle with these challenges (Table 2).

5.3.2 Semantic Segmentation on ScanNet200

Besides outdoor dataset, we have also performed experiments on indoor dataset ScanNet200 (Rozenberszki et al., 2022) with label hierarchy. ScanNet200 is an extension of the original ScanNet (Dai et al., 2017) dataset that significantly expands semantic class annotations while maintaining the original data split of 1201 training scans, 312 validation scans, and 100 test scans (1513 scans in total). The dataset contains 200 semantic classes, representing an order of magnitude increase from the original 20 classes. It was derived from ScanNet's raw semantic annotations, which initially contained 607 categories. Furthermore, a label hierarchy is provided by (Eigen & Fergus, 2015), which presents a 13-class segmentation with intermediate categories; and a fine-grained 40-class segmentation. The label hierarchy can be mapped into ScanNet200, thus we can obtain a hierarchical segmentation dataset. Five backbone models are used including SparseUNet (Graham et al., 2018), MinkowskiUNet (Choy et al., 2019), Point Transformer (PT) (Zhao et al., 2021), Point Transformer V3 (PTv3) (Wu et al., 2024), and Swin3D (Yang et al., 2023). Experimental results are given at Table 3. The results clearly demonstrate that the proposed HL can successfully improve the performance.

5.3.3 Part Segmentation on PartNet-H

We conduct further experimentation with our architecture on the task of hierarchical part segmentation. Our evaluation is based on the recently proposed large-scale PartNet dataset (Mo et al., 2019). PartNet comprises over 26,671 3D models of 24 distinct object types including 573,585 annotated part instances.

Our subsequent experiments concentrate on the three or two levels of hierarchical semantic segmentation, encompassing 17 out of the total 24 object categories present in the PartNet dataset. Moreover, to meet the label constraint of HL in Lemma 3, we reprocess the label mapping in PartNet with more details provided in the online supplementary material, and refer the resulting dataset as "PartNet-H".

Results of mIoU and OA for SparseUNet (Graham et al., 2018) are given in Tables 4 and 5, respectively. The mIoU and OA are computed for each of the 17 part categories as well as the average across three levels of segmentation: coarse- (1), middle- (2), and fine-grained (3).

The results demonstrate that HL improves the average mIoU across all 17 categories by approximately 1% to 2%. Additional results for RandLANet and PointNet++ are given in the Appendix.

5.4 Class Hierarchy Evaluation on SensatUrban-H

In this subsection, we evaluate the proposed HL approach on another 3D dataset, SensatUrban (Hu et al., 2021). This dataset is not originally annotated in a hierarchical manner. To create a hierarchical structure, we employ our class hierarchy mining method (refer to Sect. 4.4), generating a two-level class hierarchy, which is displayed by Fig. 7. The enhanced dataset, which we denote as "SensatUrban-H".

Table 3 Hierarchical semantic segmentation results on ScanNet200 (Rozenberszki et al., 2022) validation dataset. "**(+)**" and "*(−)*" stand for the positive and negative gain of metrics by HL method

Model	HL	mIoU^1	mIoU^2	mIoU^3	OA^1	OA^2	OA^3
SparseUNet (Graham et al., 2018)	w/o	73.6	52.2	24.3	84.9	83.3	79.7
	w/	74.9 **(+1.3)**	54.6 **(+2.4)**	25.1 **(+0.8)**	88.7 **(+3.8)**	83.5 **(+0.2)**	81.3 **(+1.7)**
MinkovskiUNet (Choy et al., 2019)	w/o	65.8	50.0	19.1	85.9	82.0	79.0
	w/	70.7 **(+4.9)**	51.0 **(+1.0)**	21.3 **(+2.2)**	88.3 **(+2.4)**	82.6 **(+0.6)**	79.7 **(+0.7)**
Point Transformer (Zhao et al., 2021)	w/o	72.7	53.0	23.7	88.5	83.4	80.8
	w/	73.9 **(+1.2)**	53.9 **(+0.9)**	24.1 **(+0.4)**	89.0 **(+0.5)**	83.9 **(+0.5)**	80.9 **(+0.1)**
Swin3D (Yang et al., 2023)	w/o	74.2	54.9	24.1	86.1	81.8	81.3
	w/	75.6 **(+1.4)**	56.0 **(+1.1)**	24.6 **(+0.5)**	87.5 **(+1.4)**	84.6 **(+2.8)**	81.9 **(+0.6)**
PointTransformerV3 (Wu et al., 2024)	w/o	74.1	58.5	27.1	88.4	85.1	81.7
	w/	76.6 **(+2.5)**	59.0 **(+0.5)**	27.8 **(+0.7)**	90.0 **(+1.6)**	85.4 **(+0.3)**	82.2 **(+0.5)**

Bold values indicate improvements by HL

The numerals 1, 2, and 3 correspond to the three hierarchical levels of segmentation granularity, denoting coarse-, middle-, and fine-grained categories, respectively. This notation is consistent throughout all subsequent mentions

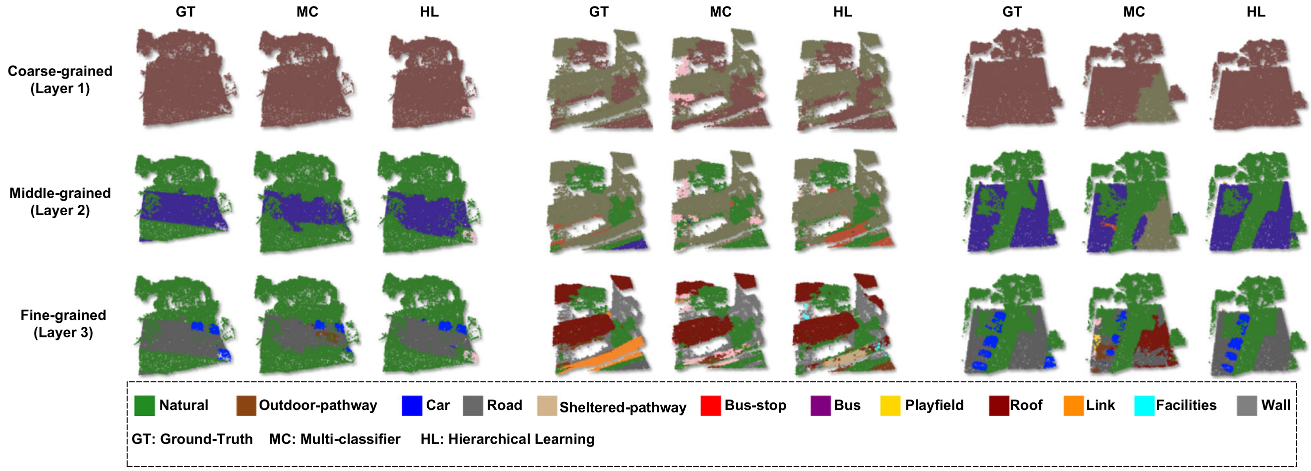


Fig. 6 Visualization of Campus3D (Li et al., 2020) semantic segmentation results. There are three groups of results and, in each group, raw pint semantic labels (GT), results of MC and HL are provided, respectively, in coarse-grained (layer 1), middle-grained (layer 2), and fine-grained (layer 3) hierarchical layer

5.4.1 Mined Class Hierarchy Quality Evaluation

In this section, we apply CH-MOS to evaluate the quality of the mined class hierarchy. For the SensatUrban-H dataset, 123 online questionnaires were distributed, and 105 responses were received. The CH-MOS values derived from these responses are presented as a boxplot in Fig. 8a. To further verify the effectiveness of the proposed class hierarchy mining method, we applied it to the leaf node class labels of the Campus3D dataset (Li et al., 2020), generated a class hierarchy, and compared it with a customized three-level hierarchy from the originally defined hierarchy (**Original**) as well as randomly generated hierarchies. A total of 100 online questionnaires were distributed, yielding 89 responses. The CH-MOS values from these responses are shown in Fig. 8b. Notably, all respondents were college students, including both graduate and undergraduate participants.

From Fig. 8, we conclude that the proposed label hierarchy mining method described in Sect. 4.4 performs effectively, producing clusters that are explainable from the perspective of human intelligence. Regarding the results for Campus3D, we observe that the better performance of our method compared to the original hierarchy is due to our hierarchy being purely language-based and thus more comprehensible for humans, whereas the original Campus3D hierarchy relies on expert domain knowledge of CityGML. This characteristic of CH-MOS reflects a potential bias of human reviewers, emphasizing the explainability of our approach.

5.4.2 Semantic Segmentation Results for Mined Class Hierarchy

Semantic segmentation results for SensatUrban-H are presented in Table 6. Aside from the OA of SparseUNet at the

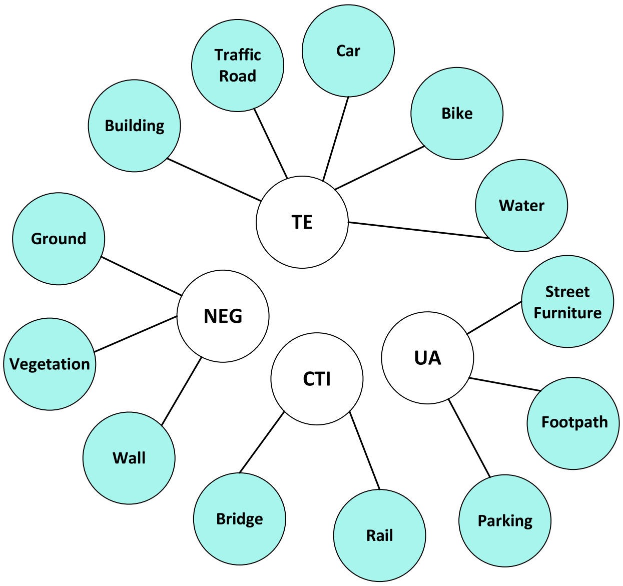


Fig. 7 Mined class hierarchy of SensatUrban-H. (NEG:Natural Elements & Ground, TE: Traffic Elements, CTI: Core Transport Infrastructure, UA: Urban Amenities)

coarse-grained level, HL has led to significant improvements. Most notably, in the fine-grained (level 2) segmentation task, HL boosts the performance of RandLANet by over 15% in terms of mIoU and around 9% in terms of OA.

5.5 Ablation Study

In order to thoroughly evaluate the impact of the CL and the HEFM component, ablation studies were performed using the Campus3D dataset. The outcomes of these studies are illustrated in Fig. 9 and detailed in Table 7. The results lead us to conclude that both CL loss and HEFM are pivotal in enhancing the coherence performance, as measured by the coherence score (κ_π), and the segmentation accuracy, as indicated by the mIOU-A and OA-A. These results reveal

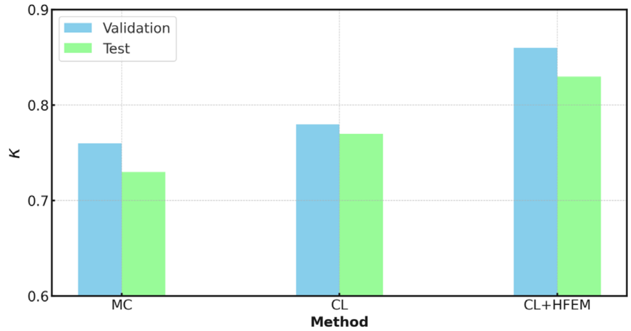
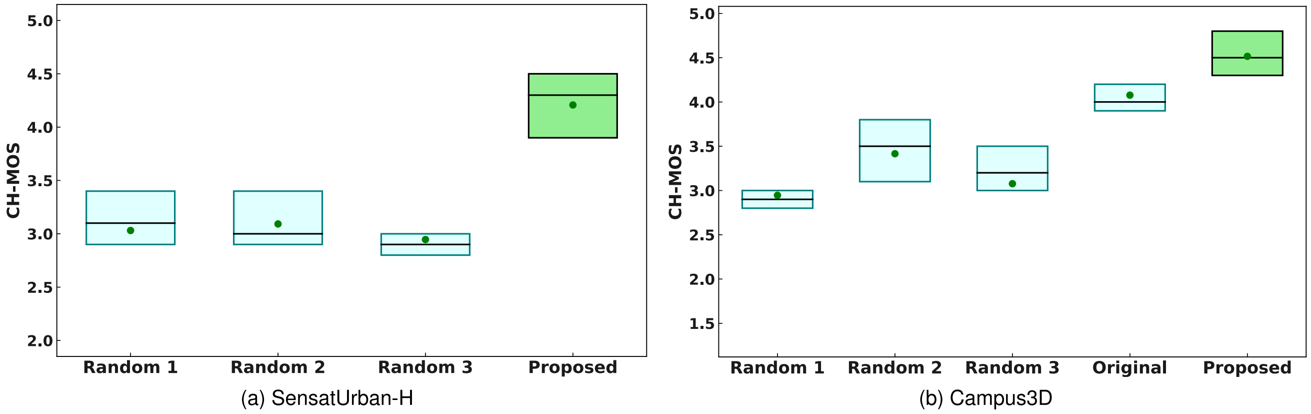


Fig. 9 κ_π values for Campus3D results on validation and test set

that the CL loss, HEFM-TF, and HEFM-BA are all essential for achieving the superior performance associated with the HL. Furthermore, an analysis that combines the insights from Fig. 9 with Tables 1 and 2 and indicates a positive correlation between solution coherence and segmentation accuracy, thereby corroborating the stated Theorem.

5.6 Computational Efficiency

The multi-head implementation in MTN introduces extra parameters and computational overhead into the original architecture. Moreover, the HEFM component which includes: 1) top-aware fusion and 2) bottom-guided aggregation. The first does not involves high computational overhead and has complexity of $O((H-1)N)$ for layer number H and point number N; as defined in equations (17) and ((18)), the second requires similarity computation among the nearest neighborhood points. The extra complexity is $O((H-1)kN)$ where $k = |\mathcal{N}|$ is the nearest neighborhood set size. This extra computation effort is minor because $(H-1)k \ll N$. Furthermore, as mentioned in Sect. 4.3, we pre-generate the neighbor indices before training thus avoid computation of pair distance calculation. We note that many models including PointNet++ and Point Transformer requires pair distance



(a) SensatUrban-H

(b) Campus3D

Fig. 8 Boxplots of CH-MOS for Class Hierarchy of SensatUrban-H and Campus3D (Li et al., 2020)

Table 4 Hierarchical semantic segmentation results of mIoU on PartNet-H dataset by SparseUNet (Graham et al., 2018)

Level (h)	Avg	Bed	Bott	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Knif	Lamp	Micro	Frid	Stor	Tab	Trash	Vase
1	–	66.3	45.6	72.1	58.6	80.2	93.2	67.0	75.5	67.3	52.6	34.1	66.6	68.6	63.2	26.8	65.1	77.3
2	–	46.5	32.1	47.9	38.9	56.3	87.1	55.0	50.6	53.5	35.1	30.4	58.6	47.3	53.0	22.2	49.1	57.7
3	–	38.0	–	43.4	–	47.1	–	45.1	–	–	–	22.5	56.0	46.9	43.5	21.2	–	–
Avg	53.1	50.3	38.9	54.5	48.7	**61.2**	90.2	55.7	63.1	**60.4**	43.9	29.0	**60.4**	54.3	**53.2**	23.4	**57.1**	67.5
1 w/ HL	–	67.1	45.6	75.8	59.0	80.3	93.7	68.2	76.5	65.4	56.8	35.0	66.6	66.5	61.9	34.1	63.2	79.2
2 w/ HL	–	48.5	40.0	47.2	39.9	55.6	87.4	55.4	49.9	53.3	37.9	31.1	56.3	47.6	53.2	29.6	46.4	59.8
3 w/ HL	–	39.8	–	42.5	–	45.2	–	47.5	–	–	–	22.2	53.2	61.62	43.0	24.8	–	–
Avg w/ HL	**54.7**	**51.8**	**42.8**	**55.1**	**49.4**	60.4	**90.6**	**57.0**	**63.2**	59.3	**47.3**	**29.5**	58.7	**58.6**	52.7	**29.5**	54.8	**69.5**

Bold values indicate the best performance

Table 5 Hierarchical semantic segmentation results of OA on PartNet-H dataset by SparseUNet (Graham et al., 2018)

Level (h)	Avg	Bed	Bott	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Knif	Lamp	Micro	Frid	Stor	Tab	Trash	Vase
1	–	83.2	95.8	80.1	91.1	93.7	97.0	75.7	92.3	85.2	65.0	74.2	99.7	99.4	86.3	92.0	82.6	97.2
2	–	60.7	95.7	81.5	72.7	83.7	96.8	75.8	74.1	78.7	60.5	72.7	92.4	91.3	79.7	77.5	75.8	94.5
3	–	57.1	–	78.9	–	83.5	–	75.6	–	–	–	68.3	91.9	90.4	79.4	70.9	–	–
Avg	82.9	67.0	95.8	80.1	81.9	**86.9**	96.9	75.7	83.2	**81.9**	62.8	71.7	**94.7**	93.7	**81.8**	80.1	**79.2**	95.8
1 w/ HL	–	84.4	95.8	93.1	91.2	92.9	97.9	76.3	93.8	85.0	69.4	75.6	99.7	99.5	85.9	96.3	81.9	97.9
2 w/ HL	–	61.5	97.3	82.3	73.6	81.4	97.7	76.0	74.6	77.7	63.9	73.9	91.9	92.5	79.3	79.1	75.1	95.7
3 w/ HL	–	58.5	–	79.4	–	83.0	–	76.2	–	–	–	69.9	91.5	91.2	79.0	71.5	–	–
Avg w/ HL	**83.8**	**68.1**	**96.6**	**84.9**	**82.4**	85.7	**97.8**	**76.2**	**84.2**	81.4	**66.7**	**73.1**	94.4	**94.4**	81.4	**82.3**	78.5	**96.8**

Bold values indicate the best performance

Table 6 Semantic segmentation results on SensatUrban-H dataset with mined class hierarchy

Model	HL	mIoU^1	mIoU^2	OA^1	OA^2
SparseUNet	w/o	50.9	51.3	86.0	85.0
	w/	**52.1**	**52.4**	85.7	**85.1**
RandLANet	w/o	43.0	43.8	53.9	62.2
	w/	**49.7**	**48.6**	**68.8**	**71.3**
PointNet++	w/o	48.1	37.9	73.7	75.1
	w/	**57.2**	**39.2**	**75.8**	**77.4**

Bold values indicate improvements by HL

The numerals 1 and 2 correspond to the hierarchical levels of segmentation granularity, representing coarse- and fine-grained categories, respectively. SparseUNet: Graham et al. (2018); RandLANet:Hu et al. (2020); PointNet++: Qi et al. (2017b)

Table 7 Ablation Study of Proposed HL Framework on Campus3D (Li et al., 2020) with SparseUNet (Graham et al., 2018). For MTN setting, original (Ori.), multi-head (MH) and multi-decoder (MD) are evaluated

MTN Setting	λ	HEFM (TF)	HEFM (BA)	mIoU-A	OA-A
Ori	–	–	–	60.2	88.0
MH	0.1	✓	✓	61.5	88.2
MD	0.01	✓	✓	60.8	88.2
MD	0.1	–	–	60.5	87.9
MD	0.1	✓	–	61.8	88.6
MD	0.1	–	✓	61.7	88.3
MD	0.1	✓	✓	**62.0**	**88.5**

Bold values indicate the best performance

TF: Top-aware Fusion; BA: Bottom-guided Aggregation

Table 8 Computational cost of proposed HL framework on Campus3D (Li et al., 2020) with SparseUNet (Graham et al., 2018) and PointTransformerv3 (Wu et al., 2024) (PTv3). For MTN setting, original (Ori.), multi-head (MH) and multi-decoder (MD) are evaluated. The number of network parameters and relative average run time per batch w.r.t original implementation are reported

Model	MTN Setting	HEFM (TF)	HEFM (BA)	# of Params	Relative Time
SparseUNet	Ori	–	–	39.2M	1.00
	MH	–	–	39.3M	1.23
	MD	–	–	55.6M	1.30
	MD	✓	–	55.7M	1.30
	MD	–	✓	55.6M	1.31
	MD	✓	✓	55.7M	1.31
PointNet++	Ori	–	–	1.0M	1.00
	MH	–	–	1.1M	1.10
	MH	✓	–	1.2M	1.10
	MH	–	✓	1.1M	1.12
	MH	✓	✓	1.2M	1.12

TF: Top-aware Fusion; BA: Bottom-guided Aggregation

calculation, thus the neighborhood query in HFEM will not induce extra cost for these models.

The comprehensive results for computational efficiency are provided in Table 8. It demonstrates that less than 25% of training time and neglectable proportion of parameters are induced by the HL method for multi-head versions of SparseUNet and PointNet++. However, the multi-decoder implementation of SparseUNet will introduce more parameters although it reaches better performance. Overall, the HL method can still improve the plain model with minor extra computation, since the computation overhead and time complexity will be H times of original model in the conventional multiple classifiers setting for H layers.

6 Conclusion

In the rapidly evolving landscape of computer vision, understanding and harnessing the inherent hierarchy of 3D objects and scenes is paramount. Through our work, we have emphasized the pivotal role of maintaining coherence across various hierarchical levels during 3D segmentation. By grounding hierarchical learning in a probabilistic context and introducing an innovative aggregation matrix, we have illuminated the intricate relationships that permeate through hierarchical structures. Our deep learning architecture, complemented by the hierarchical embedding learning module, signifies a significant step forward in this domain. Furthermore, the integration of a Large Language Model and clustering techniques to derive a hierarchical structure for detailed 3D segmentation underscores our commitment to harnessing the best of both the textual and visual worlds. The promising results from our experiments reinforce the potential of our approach, paving the way for future research and applications in this area. We encourage the academic and industrial community to delve into our publicly available source code and continue this exploration, aiming for even more refined solutions in the realm of 3D computer vision.

Appendix A Proof of Lemma 3

In order to prove Lemma 3, we first prove the following Lemma.

Lemma 4 *Given predictions of two layers $\boldsymbol{\pi}_\theta$ and $\boldsymbol{\pi}_{\theta'}$ for one sample $\boldsymbol{x}$, namely $\boldsymbol{\pi}_\theta = [p_\theta(c_1|\boldsymbol{x}), \cdots, p_\theta(c_m|\boldsymbol{x})]^\top$ and $\boldsymbol{\pi}_{\theta'} = [p_{\theta'}(c'_1|\boldsymbol{x}), \cdots, p_{\theta'}(c'_{m'}|\boldsymbol{x})]^\top$ defined on $\Omega = \{c_i\}_{i=1}^m$ and $\Omega' = \{c'_j\}_{j=1}^{m'}$ with the relationship $\Omega' \preccurlyeq \Omega$, if*

$$\kappa_{\theta,\theta'}(\boldsymbol{x}) = 1,$$

then $p_\theta(c_i|\boldsymbol{x}) = \sum_{j=1}^{m'} \mathbb{1}(c'_j \preccurlyeq c_i) p_{\theta'}(c'_j|\boldsymbol{x})$ *holds for every* i *and* $p_\theta(\cdot|\boldsymbol{x})$ *is a deterministic distribution.*

Proof. the coherent score is represented by a joint probability format given by

$$\kappa_{\theta,\theta'}(\boldsymbol{x}) = \sum_{i=1}^{m} \sum_{j=1}^{m'} \mathbb{1}(c'_j \preccurlyeq c_i) \Pr(y = c_i, y' = c'_j|\boldsymbol{x}), \quad \text{(A1)}$$

Since $p_\theta(y|\boldsymbol{x})$ and $p_{\theta'}(y'|\boldsymbol{x})$ are conditional independent when $\boldsymbol{x}$ is given,

$$\begin{aligned}\kappa_{\theta,\theta'}(\boldsymbol{x}) &= \sum_{i=1}^{m} \sum_{c'_j \preccurlyeq c_i} p_\theta(y = c_i|\boldsymbol{x}) p_{\theta'}(y' = c'_j|\boldsymbol{x}) \\ &= \sum_{i=1}^{m} p_\theta(y = c_i|\boldsymbol{x}) \sum_{c'_j \preccurlyeq c_i} p_{\theta'}(y' = c'_j|\boldsymbol{x}), \quad \text{(A2)}\end{aligned}$$

let $\hat{\pi}'_i = \sum_{c'_j \preccurlyeq c_i} p_{\theta'}(y' = c'_j|\boldsymbol{x})$ and $\pi_i = p_\theta(y = c_i|\boldsymbol{x})$ for $i = 1, \cdots, m$, we have

$$\kappa_{\theta,\theta'}(\boldsymbol{x}) = \sum_{i=1}^{m} \pi_i \hat{\pi}'_i. \quad \text{(A3)}$$

We note that $\sum_{i=1}^{m} \pi'_i = 1$ and $\pi'_i \geqslant 0$ hold for every i since the definition of $p_{\theta'}$ and Ω'. Intuitively, the largest value of $\kappa_{\theta,\theta'}$ in (A3) is 1 since the probabilistic property of $\boldsymbol{\pi}_\theta$ and $\boldsymbol{\pi}_{\theta'}$. In the following, we formally derive this conclusion and the necessary conditions of $\kappa_{\theta,\theta'} = 1$ by formulating an optimization problem based on (A3).

We suppose the optimal value of the following problem

$$\min_{\boldsymbol{\pi},\boldsymbol{\pi}'} \; -\boldsymbol{\pi}^\top \boldsymbol{\pi}' \quad \text{(A4)}$$

$$\text{s.t. } \mathbf{1}^\top \boldsymbol{\pi} = 1, \quad \text{(A5)}$$

$$\mathbf{1}^\top \boldsymbol{\pi}' = 1, \quad \text{(A6)}$$

$$\pi_i \geqslant 0, i = 1, \cdots, m, \quad \text{(A7)}$$

$$\pi'_j \geqslant 0, j = 1, \cdots, m, \quad \text{(A8)}$$

where $|\boldsymbol{\pi}| = |\boldsymbol{\pi}'| = m$. Note that $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are exactly two probability distributions when the constraints hold. To solve the problem, we define the Lagrange function of the above problem as

$$-\boldsymbol{\pi}^\top \boldsymbol{\pi}' + \lambda \mathbf{1}^\top \boldsymbol{\pi} + \lambda' \mathbf{1}^\top \boldsymbol{\pi}' + \boldsymbol{\mu}^\top \boldsymbol{\pi} + \boldsymbol{\mu}'^\top \boldsymbol{\pi}'. \quad \text{(A9)}$$

Based on the function, we consider the necessary Karush-Kuhn-Tucker (KKT) conditions of the optimal variables $\boldsymbol{\pi}'^*$, $\boldsymbol{\pi}^*$ based on the problem:

$$-\boldsymbol{\pi}'^* + \lambda \mathbf{1} = -\boldsymbol{\mu} \quad \text{(A10)}$$

$$-\boldsymbol{\pi}^* + \lambda' \mathbf{1} = -\boldsymbol{\mu}' \quad \text{(A11)}$$

$$\mu_i \pi^*_i = 0, i = 1, \cdots, m, \quad \text{(A12)}$$

$$\mu'_j \pi'^*_j = 0, j = 1, \cdots, m, \quad \text{(A13)}$$

$$\mu_i \leqslant 0, i = 1, \cdots, m, \quad \text{(A14)}$$

$$\mu'_j \leqslant 0, j = 1, \cdots, m, \quad \text{(A15)}$$

and the original constraints. Considering $I = \{i = 1, \cdots, m \mid \pi^*_i = 0\}$ and $J = \{j = 1, \cdots, m \mid \pi'^*_j = 0\}$, we have the following equations to make the KKT conditions hold,

$$|I| < m, |J| < m, \quad \text{(A16)}$$

$$\pi^*_i = \lambda' > 0, \mu_i = 0\,, \forall i \notin I, \quad \text{(A17)}$$

$$\mu_i = -\lambda', \forall i \in J, \quad \text{(A18)}$$

$$\pi'^*_j = \lambda > 0, \mu'_j = 0\,, \forall j \notin J, \quad \text{(A19)}$$

$$\mu'_j = -\lambda, \forall j \in I \quad \text{(A20)}$$

The above equations result in $I = J$, and so, $\lambda' = \lambda = 1/(m - |I|)$. The original optimization becomes,

$$\min_{m^*} -\frac{1}{m^{*2}} \text{ s.t. } m' \in \{1, \cdots, m\}, \quad \text{(A21)}$$

where $m^* = m - |I|$. The minimum value of the objective is -1 when $m^* = 1$, *i.e.*, $|I| = m - 1$. Therefore, $\boldsymbol{\pi}^* = \boldsymbol{\pi}'^*$ are the same vectors where one element is 1 and the rest of elements are 0.

As a result, when $\kappa_{\theta,\theta'}(\boldsymbol{x})$ in euqation (A3) reaches the maximum vlaue 1, we have $\pi_i = \hat{\pi}'_i$ thus $p_\theta(c_i|\boldsymbol{x}) = \sum_{j=1}^{m^*} \mathbb{1}(c'_j \preccurlyeq c_i) p_{\theta'}(c'_j|\boldsymbol{x})$ holds for every i. Moreover, $p_\theta(\cdot|\boldsymbol{x})$ is deterministic since the probability of a class is 1.

Proof of Lemma 2 When the two predictors p_θ and $p_{\theta'}$ are hierarchically coherent, because of the theoretical result in Lemma 3:

$$\sum_{c'_q \preccurlyeq c_p} p_{\theta'}(c'_q|\boldsymbol{x}) = p_\theta(c_p|\boldsymbol{x}), \quad \text{(A22)}$$

we have the fact that $p_{\theta'}(c'_q|\boldsymbol{x}) = 1$ implies $p_\theta(c_p|\boldsymbol{x}) = 1$ when $c'_q \preccurlyeq c_p$. In addition, the statement that $p_\theta(c_p|\boldsymbol{x}) = 0$ implies $p_{\theta'}(c'_q|\boldsymbol{x}) = 0$ when $c'_q \preccurlyeq c_p$ also holds.

Appendix B Proof of the Theorem

Proof Suppose we have $\Omega = \{c_p\}_{p=1}^{m}$ and $\Omega' = \{c'_q\}_{q=1}^{m'}$ as two class layers, and $\mathcal{C}$ is the super-class layer of $\mathcal{C}'$. We assume $\Omega' \preccurlyeq \Omega$. Furthermore, let $\boldsymbol{\pi}_\theta(\boldsymbol{x})$ and $\boldsymbol{\pi}_{\theta'}(\boldsymbol{x})$ stand for two predictions, $p_\theta(y|\boldsymbol{x})$ and $p_{\theta'}(y'|\boldsymbol{x})$, for a given sample

$\boldsymbol{x}$. Therefore, the coherent score is represented by a joint probability format given by

$$\kappa_{\theta,\theta'}(\boldsymbol{x}) = \sum_{p=1}^{m} \sum_{q=1}^{m'} \mathbb{1}(c'_q \preccurlyeq c_p) \mathrm{Pr}(y = c_p, y' = c'_q | \boldsymbol{x}). \tag{B23}$$

Since $p_\theta(y|\boldsymbol{x})$ and $p_{\theta'}(y'|\boldsymbol{x})$ are conditional independent when $\boldsymbol{x}$ is given, we have,

$$\begin{aligned} \kappa_{\theta_1,\theta_2}(\boldsymbol{x}) &= \sum_{p=1}^{m} \sum_{c'_q \preccurlyeq c_p} (y = c_p|\boldsymbol{x}) p_{\theta'}(y' = c'_q|\boldsymbol{x}) \\ &= \sum_{p=1}^{m} p_\theta(y = c_p|\boldsymbol{x}) \sum_{c'_q \preccurlyeq c_p} p_{\theta'}(y' = c'_q|\boldsymbol{x}) \\ &= \sum_{p=1}^{m} p_\theta^2(y = c_p|\boldsymbol{x}), \end{aligned} \tag{B24}$$

when the equation

$$p_\theta(y = c_p|\boldsymbol{x}) = \sum_{q=1}^{m'} \mathbb{1}(c'_q \preccurlyeq c_p) p_{\theta'}(y' = c'_q|\boldsymbol{x}), \tag{B25}$$

is satisfied. We note that equation (B25) can be obtained by $|\boldsymbol{\pi}_\theta(\boldsymbol{x}) - \mathbf{A}\boldsymbol{\pi}_{\theta'}(\boldsymbol{x})| = 0$ for $\mathbf{A}$ being the aggregation matrix between two class layers. Subsequently, since $\sum_{p=1}^{m} p_\theta^2(y = c_p|\boldsymbol{x})$ in equation (B24) is a Schur-convex function of $p_\theta(y|\boldsymbol{x})$ (Peajcariaac & Tong, 1992; Zhang, 1998), we have the conclusion that it decreases monotonically with the information entropy given by

$$\begin{aligned} H_\theta(y|\boldsymbol{x}) &= \sum_{y \in \Omega} p_\theta(y|\boldsymbol{x}) \log p_\theta(y|\boldsymbol{x}) && \text{(B26)} \\ &= \boldsymbol{\pi}_\theta(\boldsymbol{x})^\top \log \boldsymbol{\pi}_\theta(\boldsymbol{x}). && \text{(B27)} \end{aligned}$$

Thus, $H_\theta(y|\boldsymbol{x}) \to 0$ implies $\sum_{p=1}^{m} p_\theta^2(y = c_p|\boldsymbol{x}) \to 1$, which leads to $\kappa_{\theta_1,\theta_2}(\boldsymbol{x}) \to 1$ when $|\boldsymbol{\pi}_\theta(\boldsymbol{x}) - \mathbf{A}\boldsymbol{\pi}_{\theta'}(\boldsymbol{x})| = 0$ holds.

Acknowledgements This research was partially supported by the National Natural Science Foundation of China (Grant No. 62202395 & 62176221), and Sichuan Science and Technology Program (Grant No: 2024NSFTD0036 & 2024ZHCG0166).

Data Availability All data used in this paper are publicly available. They are: Campus3D (Li et al., 2020), ScanNet200 (Brown et al., 2020), SensatUrban (Hu et al., 2021) and PartNet (Mo et al., 2019). Based on them, we have developed their label hierarchies for our experiments, which are detailed in the supplementary document. Moreover, all source codes will be publicly available upon the acceptance of this paper.

References

Ahmed, K., Baig, M. H., & Torresani, L. (2016). Network of experts for large-scale image categorization. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14* (pp. 516–532). Springer.

Akata, Z., Reed, S., Walter, D., et al. (2015). Evaluation of output embeddings for fine-grained image classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2927–2936).

Athanasopoulos, G., Gamakumara, P., Panagiotelis, A., et al. (2020). Hierarchical Forecasting. In P. Fuleky (Ed.), *Macroeconomic Forecasting in the Era of Big Data: Theory and Practice* (Vol. 52, pp. 689–719). Cham: Springer.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., et al. (2023). Forecast reconciliation: A review. *International Journal of Forecasting, 62*, 233–258.

Audebert, N., Le Saux, B., Lefèvre, S. (2017). Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In: *Computer vision–ACCV 2016: 13th Asian conference on computer vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I 13* (pp. 180–196). Springer.

Barutcuoglu, Z., Schapire, R. E., & Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics, 22*(7), 830–836.

Bello, S. A., Yu, S., Wang, C., et al. (2020). Deep learning on 3D point clouds. *Remote Sensing, 12*(11), 1729.

Bengio, S., Weston, J., & Grangier, D. (2010). Label embedding trees for large multi-class tasks. *Advances in Neural Information Processing Systems, 23*, 489.

Bennett, P. N., & Nguyen, N. (2009). Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 11–18).

Bertinetto, L., Mueller, R., Tertikas, K., et al. (2020). Making better mistakes: Leveraging class hierarchies with deep networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12506–12515).

Bi, W., & Kwok, J. T. (2011). Multi-label classification on tree-and DAG-structured hierarchies. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 17–24).

Bi, W., & Kwok, J. T. (2015). Bayes-optimal hierarchical multilabel classification. *IEEE Transactions on Knowledge and Data Engineering, 27*(11), 2907–2918.

Bilal, A., Jourabloo, A., Ye, M., et al. (2017). Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics, 24*(1), 152–162.

Borges, H. B., & Nievola, J. C. (2012). Multi-label hierarchical classification using a competitive neural network for protein function prediction. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In *Advances in neural information processing systems* (pp. 1877–1901).

Carozza, L., Tingdahl, D., Bosché, F., et al. (2014). Markerless vision-based augmented reality for urban planning. *Computer-Aided Civil and Infrastructure Engineering, 29*(1), 2–17.

Chami, I., Gu, A., Chatziafratis, V., et al. (2020). From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems, 33*, 15065–15076.

Chen, J., Wang, P., Liu, J., et al. (2022). Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4858–4867).

Chen, T., Wu, W., Gao, Y., et al. (2018). Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In *Proceedings of the ACM international conference on multimedia* (pp. 2023–2031).

Choy, C., Gwak, J., Savarese, S. (2019). 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3075–3084).

Contributors, P. (2023). Pointcept: A codebase for point cloud perception research. https://github.com/Pointcept/Pointcept

Cui, Y., Chen, R., Chu, W., et al. (2021). Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems, 23*(2), 722–739.

Dai, A., & Nießner, M. (2018). 3D-multi-view: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European conference on computer vision* (pp. 452–468).

Dai, A., Chang, A. X., Savva, M., et al. (2017). Scannet: Richly-annotated 3D reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE.

Deng, J., Dong, W., Socher, R., et al. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Di Fonzo, T., & Girolimetto, D. (2022). Enhancements in cross-temporal forecast reconciliation, with an application to solar irradiance forecasts. arXiv preprint arXiv:2209.07146

Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision* (pp. 2650–2658).

Eisner, R., Poulin, B., Szafron, D., et al. (2005). Improving protein function prediction using the hierarchical structure of the gene ontology. In *2005 IEEE symposium on computational intelligence in bioinformatics and computational biology* (pp. 1–10). IEEE.

Fagni, T., & Sebastiani, F. (2007). On the selection of negative examples for hierarchical text categorization. In *Proc. 3rd Lang. Learn. Technol.* (pp. 24–28)

Frome, A., Corrado, G. S., Shlens, J., et al. (2013). Devise: A deep visual-semantic embedding model. *Advances in Neural Information Processing Systems, 26*, 3472.

Gao, B., Pan, Y., Li, C., et al. (2021). Are we hungry for 3D lidar data for semantic segmentation? A survey of datasets and methods. *IEEE Transactions on Intelligent Transportation Systems, 23*(7), 6063–6081.

Giunchiglia, E., & Lukasiewicz, T. (2020). Coherent hierarchical multi-label classification networks. In *Advances in neural information processing systems* (pp. 9662–9673).

Graham, B., Engelcke, M., & Van Der Maaten, L. (2018). 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9224–9232).

Guo, Y., Wang, H., Hu, Q., et al. (2020). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(12), 4338–4364.

Guo, Y., Wang, H., Hu, Q., et al. (2020). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(12), 4338–4364.

Hu, Q., Yang, B., Xie, L., et al. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11108–11117).

Hu, Q., Yang, B., Khalid, S., et al. (2021). Towards semantic segmentation of urban-scale 3d point clouds: A dataset, benchmarks and challenges. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4977–4987).

Huang, Q., Wang, W., & Neumann, U. (2018). Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2626–2635).

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and practice. OTexts.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., et al. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis, 55*(9), 2579–2589.

Jiang, L., Zhao, H., Liu., S., et al. (2019). Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the international conference on computer vision* (pp. 10433–10441).

Kaiser, D., Quek, G. L., Cichy, R. M., et al. (2019). Object vision in a structured world. *Trends in Cognitive Sciences, 23*(8), 672–685.

King, C. H., Chen, T. L., Jain, A., et al. (2010). Towards an assistive robot that autonomously performs bed baths for patient hygiene. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 319–324). IEEE.

Kobren, A., Monath, N., Krishnamurthy, A., et al. (2017). A hierarchical algorithm for extreme clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 255–264).

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML-interoperable access to 3D city models. In P. Oosterom, S. Zlatanova, & E. M. Fendel (Eds.), *Geo-information for Disaster Management* (Vol. 49, pp. 883–899). Heidelberg: Springer.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of international conference on machine learning* (pp. 170–178).

Lai, X., Liu, J., Jiang, L., et al. (2022). Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8500–8509).

Lawin, F. J., Danelljan, M., Tosteberg, P., et al. (2017). Deep projective 3D semantic segmentation. In *Computer analysis of images and patterns: 17th international conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, proceedings, Part I 17* (pp. 95–107). Springer.

Li, L., Zhou, T., Wang, W., et al. (2022). Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1246–1257).

Li, X., Li, C., Tong, Z., et al. (2020). Campus3D: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 238–246).

Li, Y., Bu, R., Sun, M., et al. (2018). PointCNN: Convolution on x-transformed points. *Advances in Neural Information Processing Systems, 31*, 3442.

Liu, J., Ni, B., Li, C., et al. (2019a). Dynamic points agglomeration for hierarchical point sets learning. In *Proceedings of the international conference on computer vision* (pp. 7546–7555).

Liu, Z., Tang, H., Lin, Y., et al. (2019b). Point-voxel CNN for efficient 3D deep learning. In *Advances in neural information processing systems*.

Long, T., & van Noord, N. (2023). Cross-modal scalable hyperbolic hierarchical clustering. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16655–16664).

Luebke, D. (2003). *Level of Detail for 3D Graphics*. Morgan Kaufmann.

Manyoky, M., Wissen Hayek, U., Heutschi, K., et al. (2014). Developing a GIS-based visual-acoustic 3d simulation for wind farm assessment. *ISPRS International Journal of Geo-Information, 3*(1), 29–48.

Mao, J., Wang, X., & Li, H. (2019). Interpolated convolutional networks for 3D point cloud understanding. In *Proceedings of the international conference on computer vision* (pp. 1578–1587).

Masera, L., & Blanzieri, E. (2019). AWX: An integrated approach to hierarchical-multilabel classification. In *Machine learning and knowledge discovery in databases: European conference, ECML PKDD, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18* (pp. 322–336). Springer.

Mirzaei, K., Arashpour, M., Asadi, E., et al. (2022). Automatic generation of structural geometric digital twins from point clouds. *Scientific Reports, 12*(1), 22321.

Mo, K., Zhu, S., Chang, AX., et al. (2019). PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 909–918).

Monath, N., Zaheer, M., Silva, D., et al. (2019). Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 714–722).

Mousavi, S. F., Safayani, M., Mirzaei, A., et al. (2017). Hierarchical graph embedding in vector space by graph pyramid. *Pattern Recognition, 61*, 245–254.

Nguyen, A., & Le, B. (2013). 3D point cloud segmentation: A survey. In *The 6th IEEE conference on robotics, automation and mechatronics (RAM)* (pp. 225–230). IEEE.

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems, 30*, 1491.

Niu, Z., Zhou, M., Wang, L., et al. (2017). Hierarchical multimodal LSTM for dense visual-semantic embedding. In *Proceedings of the international conference on computer vision* (pp. 1881–1889).

Orcutt, G. H., Watts, H. W., & Edwards, J. B. (1968). Data aggregation and information loss. *American Economic Review, 58*(4), 773–787.

Peajcariaac, J. E., & Tong, Y. L. (1992). *Convex functions, partial orderings, and statistical applications*. Academic Press.

Qi, C. R., Su, H., Mo, K., et al. (2017a). PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 652–660).

Qi, C. R., Yi, L., Su, H., et al. (2017b). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*.

Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning, PMLR* (pp. 8748–8763).

Ramaswamy, H., Tewari, A., & Agarwal, S. (2015). Convex calibrated surrogates for hierarchical classification. In *Proceedings of international conference on machine learning, PMLR* (pp. 1852–1860).

Rethage, D., Wald, J., Sturm, J., et al. (2018). Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European conference on computer vision* (pp. 596–611).

Rozenberszki, D., Litany, O., & Dai, A. (2022). Language-grounded indoor 3d semantic segmentation in the wild. In *European conference on computer vision* (pp. 125–141). Springer.

Sala, F., De Sa, C., Gu, A., et al. (2018). Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning, PMLR* (pp. 4460–4469).

Shlifer, E., & Wolff, R. W. (1979). Aggregation and proration in forecasting. *Management Science, 25*(6), 594–603.

Silla, C. N., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery, 22*, 31–72.

Su, H., Jampani, V., Sun, D., et al. (2018). SplatNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2530–2539).

Tang, H., Liu, Z., Zhao, S., et al. (2020). Searching efficient 3d architectures with sparse point-voxel convolution. In *Proceedings of the European conference on computer vision* (pp. 685–702). Springer.

Tatarchenko, M., Park, J., Koltun, V., et al. (2018). Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3887–3896).

Thomas, H., Qi, C. R., Deschaud, J. E., et al. (2019). KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the international conference on computer vision* (pp. 6411–6420).

Uyar, F., Shomstein, S., Greenberg, A. S., et al. (2016). Retinotopic information interacts with category selectivity in human ventral cortex. *Neuropsychologia, 92*, 90–106.

Vens, C., Struyf, J., Schietgat, L., et al. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning, 73*, 185–214.

Viswanathan, M., & Viswanathan, M. (2005). Measuring speech quality for text-to-speech systems: Development and assessment of a modified mean opinion score (MOS) scale. *Computer Speech & Language, 19*(1), 55–83.

Wang, W., Zhang, Z., Qi, S., et al. (2019a). Learning compositional neural information fusion for human parsing. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5703–5713).

Wang, W., Zhu, H., Dai, J., et al. (2020). Hierarchical human parsing with typed part-relation reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8929–8939).

Wang, X., Hyndman, R. J., Li, F., et al. (2022). Forecast combinations: An over 50-year review. *International Journal of Forecasting, 39*, 1518–1547.

Wang, Y., Sun, Y., Liu, Z., et al. (2019). Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (tog), 38*(5), 1–12.

Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association, 58*(301), 236–244.

Wehrmann, J., Cerri, R., & Barros, R. (2018). Hierarchical multi-label classification networks. In *Proceedings of international conference on machine learning, PMLR* (pp. 5075–5084).

Wickramasuriya, S. L., Athanasopoulos, G., & Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association, 114*(526), 804–819.

Wu, X., Jiang, L., Wang, P. S., et al. (2024). Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4840–4851).

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Transactions on Geoscience and Remote Sensing, 8*(4), 38–59.

Xu, C., & Geng, X. (2019). Hierarchical classification based on label distribution learning. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 5533–5540).

Yan, Z., Zhang, H., Piramuthu, R., et al. (2015). HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 2740–2748).

Yang, J., Fan, J., Wang, Y., et al. (2020). Hierarchical feature embedding for attribute recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13055–13064).

Yang, Y. Q., Guo, Y. X., Xiong, J. Y., et al. (2023). Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. arXiv preprint arXiv:2304.06906

Yi, L., Kim, V. G., Ceylan, D., et al. (2016). A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (ToG), 35*(6), 1–12.

Yu, F., Liu, K., Zhang, Y., et al. (2019). PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9491–9500).

Zhang, C., Li, T., & Li, C. (2023). A hierarchical reconciliation least square method for linear regression. In *Machine learning, multi agent and cyber physical systems: Proceedings of the 15th international FLINS conference (FLINS 2022)* (pp. 36–44). World Scientific.

Zhang, L., Shah, S. K., & Kakadiaris, I. A. (2017). Hierarchical multi-label classification using fully associative ensemble learning. *Pattern Recognition, 70*, 89–103.

Zhang, W., & Xiao, C. (2019). PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12436–12445).

Zhang, X. M. (1998). Optimization of Schur-convex functions. *Mathematical Inequalities & Applications, 1*(3), 319–330.

Zhang, Y., Zhou, Z., David, P., et al. (2020). PolarNet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9601–9610).

Zhao, B., Li, F., & Xing, E. (2011). Large-scale category structure aware image categorization. *Advances in Neural Information Processing Systems* 24.

Zhao, H., Jiang, L., Fu, C. W., et al. (2019). PointWeb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5565–5573).

Zhao, H., Jiang, L., Jia, J., et al. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16259–16268).

Zhu, X., Zhou, H., Wang, T., et al. (2021). Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9939–9948).

Zweig, A., Weinshall, D. (2007). Exploiting object hierarchy: Combining models from different category levels. In *Proceedings of the international conference on computer vision* (pp. 1–8). IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.