Generalized Zero-Shot Learning for Point Cloud Segmentation with Evidence-Based Dynamic Calibration

Hyeonseok Kim, Byeongkeun Kang, Yeejin Lee*

Seoul National University of Science and Technology, Republic of Korea
{khslab, byeongkeun.kang, yeejinlee}@seoultech.ac.kr

Abstract

Generalized zero-shot semantic segmentation of 3D point clouds aims to classify each point into both seen and unseen classes. A significant challenge with these models is their tendency to make biased predictions, often favoring the classes encountered during training. This problem is more pronounced in 3D applications, where the scale of the training data is typically smaller than in image-based tasks. To address this problem, we propose a novel method called E3DPC-GZSL, which reduces overconfident predictions towards seen classes without relying on separate classifiers for seen and unseen data. E3DPC-GZSL tackles the overconfidence problem by integrating an evidence-based uncertainty estimator into a classifier. This estimator is then used to adjust prediction probabilities using a dynamic calibrated stacking factor that accounts for pointwise prediction uncertainty. In addition, E3DPC-GZSL introduces a novel training strategy that improves uncertainty estimation by refining the semantic space. This is achieved by merging learnable parameters with text-derived features, thereby improving model optimization for unseen data. Extensive experiments demonstrate that the proposed approach achieves state-of-the-art performance on generalized zero-shot semantic segmentation datasets, including ScanNet v2 and S3DIS.

Introduction

Semantic segmentation of 3D point clouds refers to a task that classifies each point into a specific semantic category. Most existing methods for this task mainly use supervised learning techniques that rely on a labeled dataset where each point is already categorized (Graham, Engelcke, and van der Maaten 2018; Zhao et al. 2021). Although these supervised models perform well at segmenting categories they have been trained on, they face challenges when dealing with novel or previously unseen categories, reducing their effectiveness in real-world scenarios. This limitation arises because these methods heavily rely on labeled training data, which may not cover all possible real-world scenarios or objects.

To overcome this limitation, zero-shot learning (ZSL) provides a valuable alternative by enabling models to generalize learned knowledge to novel, previously unseen categories. However, even with zero-shot learning, achieving accurate segmentation for these novel categories remains a significant challenge. This is particularly critical in high-stakes 3D applications such as autonomous driving and medical imaging, where precise segmentation is crucial for safety and reliability.

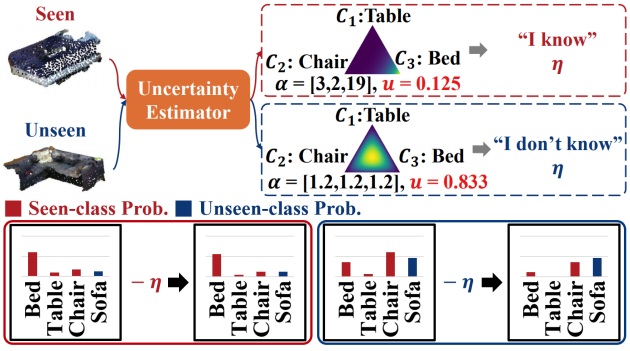


Figure 1: An illustration of E3DPC-GZSL. E3DPC-GZSL mitigates the challenge of the overconfidence problem, as shown in the bottom figures, by using a dynamic calibration (η). The upper figure shows how η is obtained from the uncertainty estimate (u), which is parameterized by the evidence (α), resulting in an adaptive calibration for both the seen (bottom left) and unseen (bottom right) datasets.

Despite its significant potential impact in various applications, research on ZSL in the 3D domain has received limited attention in the literature. Early efforts to apply ZSL to point clouds (Cheraghian, Rahman, and Petersson 2019) focused on a classification task by learning to map point cloud features into a word embedding space. Subsequent studies (Cheraghian et al. 2019) introduced an unsupervised skewness loss to address the hubness problem, which arises from the phenomenon that the nearest neighbors of many data points converge to a single hub in a high-dimensional space.

To address more practical scenarios, these techniques have been extended to generalized zero-shot learning (GZSL). GZSL seeks to recognize both seen and unseen categories during inference. There are two different configurations to achieve GZSL: transductive and inductive settings. Transductive settings allow the use of unlabeled data,

*Corresponding Author.

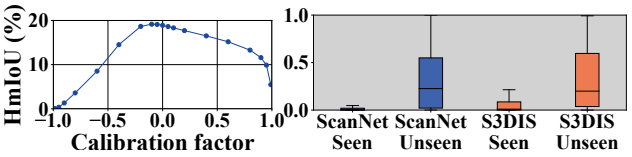
including unseen points without labels, whereas inductive settings strictly avoid using unseen points during the training phase. A critical issue with GZSL models trained in this inductive setting is that they often produce biased predictions, with a tendency to favor the seen categories used during training (Chao et al. 2016).

To overcome this problem, two main approaches are commonly used: 1) the binary classification approach and 2) the calibrated stacking approach. Methods using the binary classification approach (Zhang and Koniusz 2018; Chen et al. 2020) first use binary classification to determine whether the input data belong to seen or unseen categories. For seen categories, a supervised task is applied, while for unseen categories, a ZSL task is applied. In the calibrated stacking approach, the models (Chao et al. 2016; Michele et al. 2021; Yang et al. 2023a) adjust the final prediction probability of the GZSL model by reducing the probability of seen categories using predefined hyperparameters. By lowering the probability of seen categories, it increases the relative probability of unseen categories. However, both approaches rely heavily on hyperparameters, as demonstrated in Figure 2(a), and face the challenge of applying the same hyperparameters consistently across all input data. Figure 2(a) shows how performance varies with different predefined calibration factors, indicating that performance is dependent on them. (See also the section on uncertainty estimation).

To tackle this issue, we propose a novel method named E3DPC-GZSL, which integrates two main approaches in GZSL. First, E3DPC-GZSL enables point-wise calibrated stacking without relying on predefined hyperparameters. Unlike previous methods that apply the same stacking parameter to all samples, E3DPC-GZSL introduces a learnable calibration parameter. This parameter adjusts prediction probabilities based on the characteristics of individual samples. It is derived from estimated uncertainty levels, which act as an indicator for identifying unseen samples, but without the need for explicitly distinguishing between seen and unseen samples (See Figure 2(b)). By evaluating the prediction evidence of reliable seen data with labels, E3DPC-GZSL can implicitly distinguish between seen and unseen samples and perform dynamic calibration to the predicted class probabilities. This calibration adjusts proportionally to the uncertainty of the unseen samples – larger adjustments are made for higher uncertainties, while smaller adjustments are made for lower uncertainties. Moreover, E3DPC-GZSL introduces a new training strategy for data augmentation for unseen classes to overcome data scarcity issues. Unlike the typical visual-text space alignment in ZSL, this strategy incorporates a new approach to semantic space refinement by fusing learnable tuning parameters into text-derived features. The proposed strategy tunes text embeddings and aligns feature vectors to the refined space, resulting in a better understanding of scenes.

In summary, our contributions are as follows:

- We show that the performance of existing methods can vary when using the standard calibrated stacking approach.
- We propose a novel method, E3DPC-GZSL, which mitigates the overconfidence of zero-shot models on seen categories by redistributing prediction probabilities using estimated uncertainty.
- We propose a new strategy for tuning semantic embeddings to overcome data scarcity in 3D zero-shot learning.
- We show that the proposed method outperforms state-of-the-art (SOTA) methods for both seen and unseen classes. Extensive analysis confirms the effectiveness of our approach for generalized zero-shot semantic segmentation in 3D datasets.



(a) (b)

Figure 2: (a) Segmentation performance variation with different calibration factors on S3DIS. (b) Uncertainty difference between seen and unseen points.

Related Works

Point Cloud Semantic Segmentation. Several approaches have been proposed to accomplish semantic segmentation of 3D point clouds. One such method uses a multilayer perceptron (MLP) that processes individual points as input, including models (Qi et al. 2017b,a) specifically designed to handle unordered points directly. Another approach employs point-wise convolution techniques, which extract features by applying kernel operations to the point cloud (Hua, Tran, and Yeung 2018; Thomas et al. 2019). This includes sparse convolution methods (Graham, Engelcke, and van der Maaten 2018; Choy, Gwak, and Savarese 2019), which map the point cloud to grid cells and perform 3D convolution operations only where data exists to extract features. More recently, transformer-based methods (Zhao et al. 2021; Yang et al. 2023b) have been proposed to encode the point cloud using attention mechanisms. However, all these methods have been designed for fully supervised learning with labels. In contrast, few methods have been developed for unsupervised settings.

In the GZSL setting, a generator usually synthesizes unseen samples using semantic information, which is then used to train semantic segmentation classifiers (Michele et al. 2021; Yang et al. 2023a). Certain methods adopt a pseudo-labeling approach for the unseen data (Cheraghian et al. 2020, 2022). Additionally, some approaches incorporate extra information, such as projected 2D images (Lu et al. 2023) or geometric primitives (Chen et al. 2023), to enhance segmentation performance.

Mitigating Bias of Prediction towards Seen Categories. ZSL models typically train on samples from seen categories, resulting in a bias toward those categories. This bias is more pronounced in inductive settings, where unseen samples are not available during training. To address this issue, one approach is to distinguish samples as either seen or unseen and

then categorize them accordingly. This approach effectively breaks down the GZSL task into a supervised learning task for seen categories and a ZSL task for unseen categories. Some methods use gating networks that perform binary classification before classifying the specific category of the input data (Zhang and Koniusz 2018; Chen et al. 2020). A more recent method (Lu et al. 2024) uses multiple gating networks to classify data not only as seen or unseen, but also as ambiguous.

Another approach is to adjust the predicted probabilities or scores of a model to more accurately represent the true probability or confidence of each prediction. A simple but effective calibration technique, known as calibrated stacking, is used in point cloud semantic segmentation models (Michele et al. 2021; Yang et al. 2023a) by increasing the probabilities of unseen predictions to a specified level.

However, both approaches heavily rely on hyperparameters and face the challenge of applying the same hyperparameters consistently across all input data.

Problem Definition

ZSL Setup. Given a point cloud, a training set $\mathcal{D}_{tr}$ contains N_{tr} labeled point samples: $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{tr}}$. Each sample in the training set consists of a point $\mathbf{x}_i \in \mathbb{R}^{N_p}$ from the point cloud and its corresponding label y_i. In $\mathcal{D}_{tr}$, the labels come from a set of $\mathcal{Y}^s$ of N_s seen classes. The point $\mathbf{x}$ can comprise either 3D spatial coordinates or spatial coordinates with color components (r, g, b). Additionally, during training, class description vectors $\mathbf{t} \in \mathbb{R}^{N_t}$, either associated with N_s seen classes or N_u unseen classes, can be accompanied by y in the form of semantic attributes or natural language embeddings. Note that the experiments presented in this paper consider the inductive setting, where only class description vectors are provided and no unseen points are included.

At inference, a set of points $\mathcal{D}_{te}^u = \{\mathbf{x}_i\}_{i=1}^{N_{te}^u}$ is provided from a set $\mathcal{Y}^u$ of N_u unseen classes that is completely disjoint from $\mathcal{Y}^s$. In other words, there are no overlapping classes between $\mathcal{Y}^u$ and $\mathcal{Y}^s$, meaning $\mathcal{Y}^u \cap \mathcal{Y}^s = \emptyset$.

GZSL Setup. In the ZSL setting, samples are drawn exclusively from unseen classes during inference. However, the GZSL setting allows samples to be drawn from both seen and unseen classes. In this setting, the inference set $\mathcal{D}_{te} = \{\mathbf{x}_i\}_{i=1}^{N_{te}}$ consists of sample points from $\mathcal{Y} = \mathcal{Y}^u \cup \mathcal{Y}^s$.

Generalized Zero-Shot Semantic Segmentation. Given the label set $\mathcal{Y} = \{c_k\}_{k=1}^{N_c}$, where N_c is the total number of seen and unseen classes ($N_s + N_u$), generalized zero-shot semantic segmentation can be achieved by introducing a weight matrix $\mathbf{w}_c \in \mathbb{R}^{N_f \times N_c}$. This matrix maps an input point $\mathbf{x}$ to the output space and is obtained by using the feature vector $\mathbf{f}$ extracted by a trained encoder E, *i.e.*, $\mathbf{f} = E(\mathbf{x}) \in \mathbb{R}^{N_f}$.

Concretely, a classifier for 3D semantic segmentation can be implemented by computing class posterior probabilities p from a logit vector $\boldsymbol{\ell} = \mathbf{w}_c^T \mathbf{f}$, as follows:

$$p_k = p\left(c_k|\mathbf{x}\right) = \frac{e^{\ell_k}}{\sum_{j=1}^{N_c} e^{\ell_j}}, \tag{1}$$

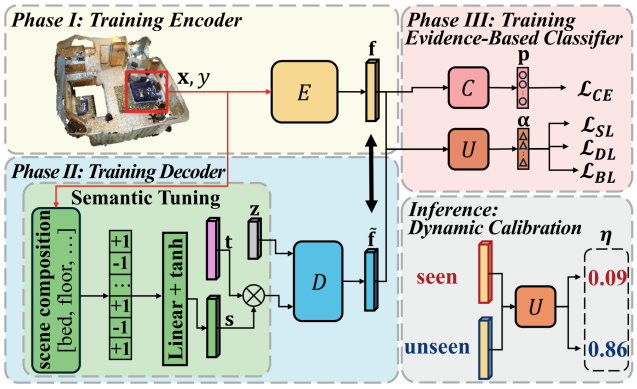


Figure 3: The E3DPC-GZSL architecture. The encoder E extracts features from the point cloud; the decoder D generates synthesized features by aligning visual features with a semantically tuned text space for training the classifier C. The classifier C then predicts the class for each point by incorporating an uncertainty estimator to adjust the predicted probabilities.

where ℓ_k and ℓ_j are the k-th and j-th elements of the vector $\boldsymbol{\ell}$, respectively, with k ranging from 1 to N_c. The predicted class for a given $\mathbf{x}$ is then computed as $\hat{y} = \underset{c_k}{\operatorname{argmax}}\, p_k$. The classifier is typically optimized using the cross-entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{N_b} \sum_{j=1}^{N_b} \sum_{c_k \in \mathcal{Y}} \mathbb{1}\left(c_k = c_{y_j}\right) p_k, \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function and N_b is the number of points in a minibatch. The cross-entropy loss function reaches its minimum when the predicted probabilities perfectly match the ground truth labels for all samples, *i.e.*, when the predicted probability for the correct class is 1, and for all other classes, it is 0. Thus, minimizing the cross-entropy loss tends to amplify the differences between the logit values, in particular by increasing the highest logit while suppressing the others. This happens because the loss function heavily penalizes incorrect classifications, forcing the model to make highly confident predictions for the most likely class (Guo et al. 2017). However, this tendency to make overly confident predictions can be problematic in GZSL. When the model encounters novel unseen classes, it may struggle to generalize effectively because it has been trained to make very sharp distinctions between the classes it has seen before.

Proposed Approach

In this section, to overcome the aforementioned issues, we propose a novel method, E3DPC-GZSL, for generalized zero-shot semantic segmentation of 3D point clouds. The overall architecture of E3DPC-GZSL is shown in Figure 3. E3DPC-GZSL consists of three main components: 1) an encoder E that extracts feature vectors $\mathbf{f}$ from input points $\mathbf{x}$, 2) a decoder D that generates synthesized features, helping to train a classifier for segmentation and transfer knowledge

from seen to unseen categories, and 3) a classifier C with an uncertainty estimator U that performs segmentation by using the feature vectors produced by E and D. Note that during inference, only the feature encoder E, the classifier C, and the uncertainty estimator U are used, while the decoder, which assisted the classifier in learning potential unseen features, is detached.

The training process of E3DPC-GZSL is divided into three phases. Below, we describe each phase and outline our strategies for achieving reliable predictions.

Phase I: Training Encoder

This training process aims to optimize the encoder E to extract a distinctive feature vector $\mathbf{f}$ from an input $\mathbf{x}$ for category identification. The knowledge gained from E is then used to train the decoder D for feature vector synthesis. The feature vectors produced by D are then used to learn the parameters for C and U.

In this phase, the encoder E is trained from scratch by optimizing the cross-entropy loss over seen samples of $\mathcal{D}_{tr}$. This approach is taken for two main reasons: first, there are no effective pre-trained foundation models for 3D point clouds; second, the configuration of 3D datasets varies significantly from task to task. For instance, in classification datasets (Wu et al. 2015; Uy et al. 2019), each point cloud sample typically contains a single object without background, whereas segmentation and detection datasets (Dai et al. 2017; Armeni et al. 2017) contain multiple objects with background in point cloud samples.

Phase II: Training Decoder

This training process has two goals: 1) to overcome data scarcity by training the decoder D to enable feature synthesis, and 2) to improve feature representation for synthesis by conditioning on both scene- and point-wise semantics.

Given a point $\mathbf{x}$ of the class c_y from $\mathcal{D}_{tr}$, the pre-trained encoder E extracts the feature vector $\mathbf{f}$. As described in Figure 3, the decoder D is then trained to generate the feature vectors $\tilde{\mathbf{f}}$ from a vector $\mathbf{z} \in \mathbb{R}^{N_z}$. The vector $\mathbf{z}$ is randomly sampled from a uniform distribution in the range $[0, 1)$. To mitigate data scarcity issues in zero-shot learning, D is conditioned on auxiliary prior knowledge about the class in the form of a text-driven feature embedding $\mathbf{t}$, which represents the class for feature synthesis.

Semantic Tuning. In E3DPC-GZSL, the text embedding $\mathbf{t}$ is tuned by introducing a learnable feature vector derived from the scene semantics, denoted as $\mathbf{s} \in \mathbb{R}^{N_t}$. This vector is constructed based on a scene composition descriptor in $\mathbb{R}^{N_c}$, whose elements are either 1 or -1. A value of 1 indicates the presence of a class in the scene, while a value of -1 indicates the absence of objects belonging to that class.

To learn $\mathbf{s}$, the scene composition descriptor is passed through a fully connected layer with a hyperbolic tangent activation function. This introduction of $\mathbf{s}$ has two key advantages: it improves the quality of synthesized features by exploiting both global scene-based and local point-based conditions, and it acts similarly to adjusting prompts that include a word for a class name (*e.g.*, [chair]) to prompts that include the class name with the scene description (*e.g.*, [chair] near [table] in the room). Once $\mathbf{s}$ is obtained, $\mathbf{t}$ is tuned by adding point by point. The synthesized feature vector $\tilde{\mathbf{f}}$ is then produced from $\mathbf{z}$ and $\mathbf{t} \otimes \mathbf{s}$, *i.e.*, $\tilde{\mathbf{f}} = D(\mathbf{z}, \mathbf{t} \otimes \mathbf{s})$, where $\otimes$ denotes an point-wise multiplication operation.

Opimization. The decoder D is trained by following the approach in (Yang et al. 2023a), minimizing the decoder loss. This decoder loss consists of the discrepancy loss between $\mathbf{f}$ and $\tilde{\mathbf{f}}$, the contrastive loss between positive and negative pairs, as well as the prototype distance loss across classes. Once the parameters of D are learned, the synthesized features align with the semantic space derived from tunable text representations.

Phase III: Training Evidence-Based Classifier for 3D Semantic Segmentation

The goal of this training process is twofold: first, to obtain the parameters of a classifier C for 3D semantic segmentation, and second, to learn parameters of U for evaluating prediction confidence.

Learning Segmentation. As shown in Figure 3, to learn the parameters of C, the pre-trained encoder E is used to encode the samples of seen classes, while the pre-trained decoder D generates synthetic features for unseen classes. C is then optimized using the cross-entropy loss on feature vectors produced by E and D, following the standard framework outlined in (1) and (2). Here, the resulting class probabilities are represented as a vector $\boldsymbol{p} \in \mathbb{R}^{N_c}$, where p_k denotes the probability of the class c_k.

Uncertainty Estimation. To avoid overconfident prediction due to prior knowledge of seen classes, at inference, the calibrated stacking method (Chao et al. 2016) is used to compute class probabilities, resulting in the vector $\boldsymbol{p'} \in [0, 1]^{N_c}$:

$$p'_k = p_k - \eta \cdot \mathbb{1}_{\mathcal{Y}^s}(c_k), \tag{3}$$

where p'_k represents the probability of the class c_k; $\mathbb{1}_A(x)$ is denoted as $\mathbb{1}(x \in A)$, *i.e.*, $\mathbb{1}(c_k \in \mathcal{Y}^s)$; and the prediction probability is redistributed using a calibration factor η in the range $[0, 1]$. In (3), η is used to decrease the probabilities for seen classes, which consequently increases the relative probabilities of unseen classes. Once the class probabilities are computed, the predicted class is assigned as $\hat{y} = \underset{c_k}{\operatorname{argmax}} p'_k$. However, segmentation performance can degrade when applying a pre-defined η to (3), as demonstrated in Figure 2(a). To mitigate this performance degradation, η is treated as an adjustable parameter in E3DPC-GZSL. It is learned from uncertainty estimation derived based on evidence theory (Shafer 1976) and subjective logic (Jø sang 2016).

In standard uncertainty estimation (Sensoy, Kaplan, and Kandemir 2018; Ulmer, Hardmeier, and Frellsen 2023; Wang et al. 2024; Zong et al. 2024), the Dirichlet distribution, characterized by a concentration parameter vector $\boldsymbol{\alpha}$, is used as a conjugate prior for Bayesian inference. When the Dirichlet distribution is uniform, it acts as a non-informative prior, representing complete uncertainty about the outcomes. For example, when minimal prior knowledge is assigned to all concentration parameters relevant to class probability (*e.g.*, $\alpha_k = 1$ for all k classes, where $\alpha \geq 1$), it implies

no preference for any particular class (See Figure 1). This reflects maximum uncertainty and a lack of evidence, meaning there is no bias or skew toward the probability of any particular class.

This uncertainty (or, conversely, evidence) can be modeled using evidence theory (Shafer 1976; Sensoy, Kaplan, and Kandemir 2018), where the degree of uncertainty is inversely proportional to the total amount of evidence represented by α with the relationship $\alpha = evidence + 1$:

$$u = \frac{K}{\alpha_0}, \tag{4}$$

where α_0 is defined as the sum of all concentration parameters, *i.e.*, $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ and K denotes the number of the parameter α.

To derive η based on prediction uncertainty, the uncertainty is measured exclusively using reliable sets of seen labels with $K = N_s$, considering two scenarios: 1) For unseen classes, u ideally reaches its maximum by accumulating the evidence (α) from seen classes. This occurs when all α are associated with seen classes, their class probabilities have minimal evidence. As a result, given unseen samples, high uncertainties associated with seen classes result in low class probabilities and a large η, thereby redistributing class probabilities more evenly for unseen classes. 2) For seen classes with labels, a skewed distribution towards a given class, driven by low uncertainty, increases prediction evidence with a small η. Based on this reasoning, the process for estimating η is as follows: model the evidence distribution as Dirichlet, estimate its concentration parameters α (substituting the evidence), compute the uncertainty u from the estimated α, and then estimate η from u (See Figure 1).

Learning η. The module U for estimating α is optimized on the expected probability of evidence π_k for each class, defined as $\pi_k = \alpha_k/\alpha_0$. Three objectives guide it: first, to estimate parameters that minimize Bayesian risk to improve evidence for class prediction ($\mathcal{L}_{SL}$); second, to regularize the optimization process ensuring balanced prediction for seen and unseen classes ($\mathcal{L}_{DL}$); and third, to improve evidence estimates by identifying unseen classes from seen ones ($\mathcal{L}_{BL}$):

$$\mathcal{L}_{EV} = \mathcal{L}_{SL} + \lambda_{DL}\mathcal{L}_{DL} + \lambda_{BL}\mathcal{L}_{BL}, \tag{5}$$

where λ_{DL} and λ_{BL} are the weighting coefficients.

The **segmentation loss (SL)** is the posterior expected loss designed to minimize Bayesian risk using the cross-entropy loss function. Assuming a Dirichlet conjugate prior, the loss is formulated as follows (for a detailed derivation, refer to Appendix A):

$$\begin{aligned}
&\mathcal{L}_{SL} \\
&= \frac{1}{N_b'} \sum_{j=1}^{N_b'} \int \frac{1}{B(\boldsymbol{\alpha_j})} \prod_{k'=1}^{N_s} \pi_{j,k'}^{\alpha_{j,k'}-1} \Big[- \sum_{k=1}^{N_s} \mathbb{1}\big(c_k = c_{y_j}\big) \log \pi_{j,k} \Big] d\boldsymbol{\pi} \\
&= -\frac{1}{N_b'} \sum_{j=1}^{N_b'} \sum_{k=1}^{N_s} \mathbb{1}\left(c_k = c_{y_j}\right) \left(\psi\left(\alpha_{j,k}\right) - \psi\left(\alpha_{j,0}\right)\right),
\end{aligned} \tag{6}$$

where $B(\cdot)$ represents the multivariate beta function; $\Gamma(\cdot)$ denotes the gamma function, and $\psi(\cdot)$ is the digamma function, defined as $\psi(x) = \frac{d}{dx} \log \Gamma(x)$; and N_b' is the number of seen samples in a minibatch.

The **divergence loss (DL)** regularizes the model by reducing the Kullback-Leibler divergence from a uniform Dirichlet distribution to ensure reliable uncertainty prediction. A modified vector $\tilde{\alpha}$ is introduced to avoid misleading evidence when labeled samples are provided and to ensure high uncertainty for unseen ones, similar to (Sensoy, Kaplan, and Kandemir 2018):

$$\begin{aligned}
\tilde{\boldsymbol{\alpha}} = \mathbb{1}_{\mathcal{Y}^s}\left(c_y\right) \cdot \Big\{ \boldsymbol{y} \otimes \big(\mathbf{1} + \tfrac{1}{\sqrt{\boldsymbol{\alpha}}}\big) + (\mathbf{1} - \boldsymbol{y}) \otimes \sqrt{\boldsymbol{\alpha}} \Big\} \\
+ \mathbb{1}_{\mathcal{Y}^u}\left(c_y\right) \cdot \boldsymbol{\alpha},
\end{aligned} \tag{7}$$

where $\mathbf{1} \in \mathbb{R}^{N_s}$ is the vector of ones, $\otimes$ denotes element-wise multiplication, and $\boldsymbol{y}$ is provided in a one-hot encoded format. Using $\tilde{\boldsymbol{\alpha}}$, the module estimates the evidence for each class by preventing incorrect predictions for unseen samples from being biased towards any seen classes (see Appendix A for full derivation):

$$\begin{aligned}
\mathcal{L}_{DL} &= \frac{1}{N_b} \sum_{j=1}^{N_b} \mathrm{KL}\left[Dir\left(\boldsymbol{\pi}_j | \tilde{\boldsymbol{\alpha}}_j\right) \middle\| Dir\left(\boldsymbol{\pi}_j | \mathbf{1}\right) \right] \\
&= \frac{1}{N_b} \sum_{j=1}^{N_b} \Big(\log \frac{1}{\Gamma(N_s) B(\tilde{\boldsymbol{\alpha}}_j)} \\
&\qquad + \sum_{k=1}^{N_s} (\tilde{\alpha}_{j,k} - 1)\big(\psi(\tilde{\alpha}_{j,k}) - \psi(\alpha_{j,0})\big)\Big).
\end{aligned} \tag{8}$$

In addition to the loss functions, the **binary loss (BL)** is introduced to ensure that the evidence is not biased toward any class when handling unseen samples. The loss function reduces uncertainty for seen samples and simultaneously increases uncertainty for unseen samples, encouraging balanced predictions:

$$\begin{aligned}
\mathcal{L}_{BL} = -\frac{1}{N_b} \sum_{j=1}^{N_b} \big[&\mathbb{1}_{\mathcal{Y}^s}\left(c_{y_j}\right) \log u_j \\
&+ \mathbb{1}_{\mathcal{Y}^u}\left(c_{y_j}\right) \log(1 - u_j) \big].
\end{aligned} \tag{9}$$

In (9), by reducing uncertainty for seen samples, the module gains confidence in its predictions for seen classes, leading to more decisive and accurate results. In contrast, by increasing uncertainty for unseen samples, the module avoids making overly confident predictions about unseen samples. This increase in uncertainty prevents overfitting and ensures that the module recognizes its limitations when it encounters new inputs.

Inference. Using the learned parameters of U to estimate α, the dynamic calibration factor is defined as $\eta = u - \bar{u}$, where $\bar{u}$ is defined as the average estimated uncertainty of unseen samples predicted from C before applying calibrated stacking.

Experiments

In this section, we provide details on various experiments designed to evaluate E3DPC-GZSL. Note that additional results and analyses are provided in the supplementary material.

	Training set		ScanNet v2				S3DIS			
	Encoder	Classifier	mIoU			HmIoU	mIoU			HmIoU
			Seen	Unseen	All		Seen	Unseen	All	
Full supervision	$\mathcal{Y}^s \cup \mathcal{Y}^u$	$\mathcal{Y}^s \cup \mathcal{Y}^u$	43.3	51.9	45.1	47.2	74.0	50.0	66.6	59.6
Full supervision only for classifier	$\mathcal{Y}^s$	$\mathcal{Y}^s \cup \mathcal{Y}^u$	41.5	39.2	40.3	40.3	60.9	21.5	48.7	31.8
Supervision with seen	$\mathcal{Y}^s$	$\mathcal{Y}^s$	39.0	0.0	31.3	0.0	70.2	0.0	48.6	0.0
3DGenZ (Michele et al. 2021)	$\mathcal{Y}^s$	$\mathcal{Y}^s \cup \mathcal{Y}^{\tilde{u}}$	32.8	7.7	27.8	12.5	53.1	7.3	39.0	12.9
3DPC-GZSL (Yang et al. 2023a)	$\mathcal{Y}^s$	$\mathcal{Y}^s \cup \mathcal{Y}^{\tilde{u}}$	34.5	14.3	30.4	20.2	58.9	9.7	43.8	16.7
E3DPC-GZSL (ours)	$\mathcal{Y}^s$	$\mathcal{Y}^s \cup \mathcal{Y}^{\tilde{u}}$	**36.1**	**15.4**	**32.0**	**21.6**	**67.9**	**12.0**	**50.7**	**20.4**

Table 1: Performance Comparisons of 3D GZSL semantic segmentation benchmarks in terms of mIoU(%) and HmIoU(%).

Experimental Setup

Datasets. We evaluate the proposed E3DPC-GZSL method using the S3DIS dataset (Armeni et al. 2017) and the ScanNet v2 dataset (Dai et al. 2017), following the data splitting protocol outlined in previous studies (Michele et al. 2021; Yang et al. 2023a).

The ScanNet v2 dataset, collected indoors, consists of $1,201$ point cloud scenes for training and 312 point cloud scenes for evaluation. The training scenes encompass sixteen seen classes ($N_s = 16$), while the evaluation scenes include four unseen classes (desk, bookshelf, sofa, and toilet, $N_u = 4$) and sixteen seen classes.

The S3DIS dataset consists of 272 scenes from 6 indoor areas, covering 13 classes. Areas 2, 3, 4, 5, and 6 (228 scenes) are used for training with nine seen classes ($N_s = 9$), while Area 1 with 44 scenes is designated as the evaluation dataset. This evaluation set includes four unseen classes (beam, column, window, and sofa, $N_u = 4$) and nine seen classes.

Evaluation Metrics. The performance of E3DPC-GZSL is assessed using two metrics. The first metric is the mean Intersection-over-Union (mIoU), which measures the average overlap between the predicted and ground truth segmentations across all classes. We assess three types of mIoU: for seen classes, unseen classes, and a combined measure for both. The second metric used is the Harmonic mean of mIoU (HmIoU), defined as $2 \times (\text{mIoU}(\mathcal{Y}^s) \times \text{mIoU}(\mathcal{Y}^u)) \,/\, (\text{mIoU}(\mathcal{Y}^s) + \text{mIoU}(\mathcal{Y}^u))$.

Implementation Details. We implement the proposed method using the PyTorch framework. Following the 3D GZSL semantic segmentation setting (Michele et al. 2021), the sample points contain spatial coordinates for the ScanNet v2 ($N_p = 3$) and spatial coordinates plus color information for the S3DIS ($N_p = 6$). The experiments use the FKAConv (Boulch, Puy, and Marlet 2020) network on the ScanNet v2 and the ConvPoint (Boulch 2020) network on the S3DIS as the encoder and classifier. The encoder E encodes the inputs into 64-dimensional feature vectors ($N_f = 64$) for ScanNet v2 and 128 for S3DIS. The decoder D used is the generative moment matching network (GMMN) (Li, Swersky, and Zemel 2015). We employ 600-dimensional ($N_t = 600$) text embeddings, which are a concatenation of GloVe (Pennington, Socher, and Manning 2014) and Word2Vec (Mikolov et al. 2013), as class description vectors. The proposed model is trained with a learning rate of $7e-2$, a batch size of 4, and a poly learning rate scheduler with a base of 0.9 and 30 epochs. The Adam optimizer is used for ScanNet v2, while the SGD optimizer is used for S3DIS. Each minibatch contains 8192 sample points ($N_b = 8192$). λ_{DL} and λ_{BL} are set to 0.005 and 0.01 for ScanNet v2 and 0.005 and 0.1 for S3DIS, respectively. The uncertainty estimator utilizes the same network as the classifier, employing the exponential function as its activation function.

Experimental Results

The effectiveness of our proposed methods is compared with SOTA methods in the inductive generalized zero-shot learning setting, where the encoder is trained with seen data ($\mathcal{Y}^s$) and the classifier is trained with both seen and augmented unseen data $\mathcal{Y}^{\tilde{u}}$. The methods compared are 3DGenZ (Michele et al. 2021) and 3DPC-GZSL (Yang et al. 2023a). The results of this comparison are summarized in Table 1. Additionally, for reference, we provide performance for three different supervised settings using the same encoder and classifier as the zero-shot models. Our proposed method achieves improvements over the state-of-the-art method in both seen and unseen mIoU metrics. Specifically, it demonstrates an increase of 1.4% HmIoU on the ScanNet v2 and 3.7% HmIoU on the S3DIS. Note that qualitative results and additional evaluations on outdoor benchmarks are provided in the supplementary material.

Further Discussion

Analysis of Component Effectiveness. The effectiveness of the proposed semantic tuning and the uncertainty estimator is validated by removing and adding each component. The analysis results are summarized in Table 2. In the table, "B" represents the baseline consisting of the encoder without calibrated stacking ($\eta = 0$), decoder without semantic tuning, and standard classifier; "S" represents the addition of the semantically tuned decoder; and "U" represents the addition of the dynamically calibrated classifier. It is evident that dynamic calibration using module U effectively adjusts the module's predictions independent of semantic tuning. Moreover, when semantic tuning is applied, it enhances classifier performance by increasing the expressive power of the decoder.

Dataset	B	S	U	mIoU			HmIoU
				Seen	Unseen	All	
ScanNet v2	✓	–	–	34.78	14.80	30.79	20.77
	✓	–	✓	34.86	14.87	30.87	20.85
	✓	✓	–	36.09	15.23	31.91	21.42
	✓	✓	✓	**36.11**	**15.40**	**31.97**	**21.59**
S3DIS	✓	–	–	65.03	10.17	48.15	17.60
	✓	–	✓	66.58	11.27	49.56	19.28
	✓	✓	–	66.46	11.02	49.40	18.90
	✓	✓	✓	**67.90**	**12.01**	**50.70**	**20.42**

Table 2: Analysis of the effects of each module on segmentation performance.

Dataset	$\mathcal{L}_{SL}$	$\mathcal{L}_{DL}$	$\mathcal{L}_{BL}$	mIoU			HmIoU
				Seen	unseen	All	
ScanNet v2	–	–	–	36.09	15.23	31.91	21.42
	✓	–	–	**36.14**	15.31	**31.98**	21.51
	✓	✓	–	36.13	15.33	31.97	21.53
	✓	–	✓	**36.14**	15.32	**31.98**	21.51
	✓	✓	✓	36.11	**15.40**	31.97	**21.59**
S3DIS	–	–	–	66.46	11.02	49.40	18.90
	✓	–	–	66.64	11.19	49.58	19.17
	✓	✓	–	67.19	11.66	50.10	19.89
	✓	–	✓	66.84	11.39	49.78	19.46
	✓	✓	✓	**67.90**	**12.01**	**50.70**	**20.42**

Table 3: Ablation study on loss functions for uncertainty estimation.

Analysis of Uncertainty Estimator. In addition, to further analyze the proposed uncertainty estimation module, we evaluate the effectiveness of each loss function. The analysis results are presented in Table 3. For reference, the performance of the B+S model in Table 2 is reported in the first row for each dataset as the baseline. As expected, segmentation accuracy on unseen data improves by redistributing the model's prediction probabilities, which coincides with HmIoU improvements, reflecting enhanced performance on unseen data.

For the S3DIS dataset, the results clearly demonstrate the effectiveness of the uncertainty estimator for unseen data. However, its impact on ScanNet v2 is less pronounced, showing only marginal improvement. To explore this further, we analyze and measure the confidence histograms and reliability diagrams of the models without applying calibrated stacking. The confidence histograms in the top row of Figure 4 show the distribution of prediction confidence, represented as softmax probabilities of the classifier C associated with the predicted class label. The reliability diagrams in the bottom row of Figure 4 show accuracy as a function of confidence, calculating the accuracy of each bin in the confidence histograms (*i.e.*, the ratio of correct predictions of each bin). If the model produces balanced outputs, the diagram would show the identity function labeled

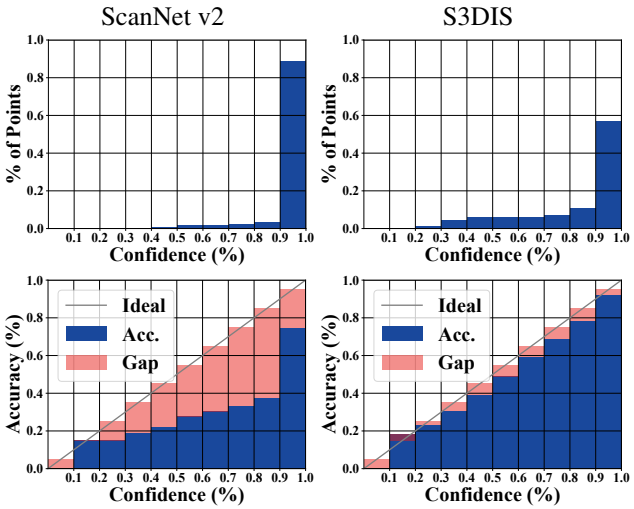


Figure 4: Analysis of model confidence on ScanNet v2 and S3DIS: confidence histograms (top) and reliability diagrams (bottom). The Gap represents the difference between the ideal accuracy and the measured accuracy.

"ideal" in the graph. Any deviation from this identity function indicates that the model produces overconfident outputs, with one class of probability significantly higher than the others (Guo et al. 2017). As evidenced in the confidence histogram for ScanNet v2, the model tends to produce outputs with a significantly high probability of one class. This overconfidence limits the effectiveness of calibrated stacking, which is intended to reduce the bias toward the classes seen. Since the probabilities are already very high, subtracting η does not effectively shift the peak of the probability distribution. In contrast, the model confidence in the outputs for S3DIS is relatively lower compared to ScanNet v2. Consequently, the effect of calibrated stacking is much more pronounced for S3DIS, resulting in a more effective redistribution of the probability distribution.

Conclusion

We proposed E3DPC-GZSL, a novel approach for generalized zero-shot point cloud semantic segmentation. Our method exploits the uncertainty of input points to dynamically calibrate classifier predictions. This uncertainty-based strategy helps to mitigate the bias of zero-shot models towards seen classes, thereby improving the generalization performance of the model. Additionally, to address the issue of data scarcity, we introduced a novel training strategy that refines the semantic space by applying semantic tuning to text embeddings. Our experiments show that E3DPC-GZSL outperforms SOTA methods in 3D semantic segmentation. Despite the significant performance improvements over SOTA, the impact is less pronounced on models that tend to produce overconfident results with high probabilities. Regularizing the model's overconfidence with biased predictions in a zero-shot setting could improve performance in such cases. We leave this as a direction for future research.

Acknowledgments

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean government (MSIT), 3D Digital Media Streaming Service Technology, under Grant No. RS-2023-00229330; and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1062950).

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2017. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543.

Boulch, A. 2020. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88: 24–34.

Boulch, A.; Puy, G.; and Marlet, R. 2020. FKAConv: Feature-Kernel Alignment for Point Cloud Convolution. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Chao, W.-L.; Changpinyo, S.; Gong, B.; and Sha, F. 2016. An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild. In *European Conference on Computer Vision (ECCV)*, 52–68.

Chen, R.; Zhu, X.; Chen, N.; Li, W.; Ma, Y.; Yang, R.; and Wang, W. 2023. Bridging Language and Geometric Primitives for Zero-shot Point Cloud Segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 5380–5388.

Chen, X.; Lan, X.; Sun, F.; and Zheng, N. 2020. A Boundary Based Out-of-Distribution Classifier for Generalized Zero-Shot Learning. In *European Conference on Computer Vision (ECCV)*, 572–588.

Cheraghian, A.; Rahman, S.; Campbell, D.; and Petersson, L. 2019. Mitigating the Hubness Problem for Zero-Shot Learning of 3D Objects. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, 41.

Cheraghian, A.; Rahman, S.; Campbell, D.; and Petersson, L. 2020. Transductive Zero-Shot Learning for 3D Point Cloud Classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 912–922.

Cheraghian, A.; Rahman, S.; Chowdhury, T. F.; Campbell, D.; and Petersson, L. 2022. Zero-Shot Learning on 3D Point Cloud Objects and Beyond. *International Journal of Computer Vision*, 130: 2364–2384.

Cheraghian, A.; Rahman, S.; and Petersson, L. 2019. Zero-shot Learning of 3D Point Cloud Objects. In *2019 16th International Conference on Machine Vision Applications (MVA)*, 1–6.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3075–3084.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2432–2443.

Graham, B.; Engelcke, M.; and van der Maaten, L. 2018. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9224–9232.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 1321–1330.

Hua, B.-S.; Tran, M.-K.; and Yeung, S.-K. 2018. Pointwise Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 984–993.

Jø sang, A. 2016. Generalising Bayes' theorem in subjective logic. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 462–469. IEEE.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *Proceedings of the 32th International Conference on Machine Learning (ICML)*, 1718–1727.

Lu, Y.; Jiang, Q.; Chen, R.; Hou, Y.; Zhu, X.; and Ma, Y. 2023. See More and Know More: Zero-shot Point Cloud Segmentation via Multi-modal Visual Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 21674–21684.

Lu, Z.; Lu, Z.-M.; Yu, Y.; He, Z.; Luo, H.; and Zheng, Y. 2024. Learning Multiple Criteria Calibration for Generalized Zero-shot Learning. *Knowledge-Based Systems*, 300: 112131.

Michele, B.; Boulch, A.; Puy, G.; Bucher, M.; and Marlet, R. 2021. Generative Zero-Shot Learning for Semantic Segmentation of 3D Point Clouds. In *2021 International Conference on 3D Vision (3DV)*, 992–1002.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, volume 30.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017b. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660.

Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential Deep Learning to Quantify Classification Uncertainty. In *Advances in Neural Information Processing Systems*.

Shafer, G. 1976. *A mathematical theory of evidence*, volume 42. Princeton university press.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6411–6420.

Ulmer, D.; Hardmeier, C.; and Frellsen, J. 2023. Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods For Uncertainty Estimation. arXiv:2110.03051.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1588–1597.

Wang, R.; Zhao, R.-W.; Zhang, X.; and Feng, R. 2024. Towards Evidential and Class Separable Open Set Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 5572–5580.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1912–1920.

Yang, Y.; Hayat, M.; Jin, Z.; Zhu, H.; and Lei, Y. 2023a. Zero-Shot Point Cloud Segmentation by Semantic-Visual Aware Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11586–11596.

Yang, Y.-Q.; Guo, Y.-X.; Xiong, J.-Y.; Liu, Y.; Pan, H.; Wang, P.-S.; Tong, X.; and Guo, B. 2023b. Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding. arXiv:2304.06906.

Zhang, H.; and Koniusz, P. 2018. Model Selection for Generalized Zero-shot Learning. In *European Conference on Computer Vision Workshops (ECCVW)*, 198–204.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16259–16268.

Zong, C.-C.; Wang, Y.-W.; Xie, M.-K.; and Huang, S.-J. 2024. Dirichlet-Based Prediction Calibration for Learning with Noisy Labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 17254–17262.