The supplementary materials are organized as follows. Appendix **A** provides the background knowledge on the Dirichlet distribution. In Appendix **B** we review the architecture of the graph posterior network (GPN) [32] together with our discussion on oversight of [32, Theorem 1]. Appendix **C** details the proofs of all the theorems and corollaries discussed in the main paper. We provide detailed descriptions of baseline models, datasets, and hyperparameter tuning for the experiments in Appendix **D**. Lastly, Appendix **E** includes more experimental results that we are unable to fit into the paper.

A Dirichlet Distribution

A non-degenerate Dirichlet distribution, denoted by $\mathrm{Dir}(\boldsymbol{\alpha})$, is parameterized by the concentration parameters $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_K]^\intercal$ with $\alpha_k > 1$ for $k \in [K]$. More specifically, the Dirichlet distribution with parameters $\alpha_1, \cdots, \alpha_K$ has a probability density function (pdf) given by

$$\mathrm{pdf}(\mathbf{p}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} p_k^{\alpha_k - 1}, \tag{12}$$

where $\{p_k\}_{k=1}^K$ belongs to the standard probability simplex, thus $\sum_{k=1}^K p_k = 1$ and $p_k \in [0, 1], \forall k \in [K]$, and the normalizing constant $B(\boldsymbol{\alpha})$ is expressed in terms of the Gamma function $\Gamma(\cdot)$, i.e.,

$$B(\boldsymbol{\alpha}) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma\left(\Sigma_k \alpha_k\right)}. \tag{13}$$

Under the semi-supervised learning setting, a set of labels is available, denoted by $\mathbb{L} \subset \mathcal{V}$. For $i \in \mathbb{L}$, the class label $y_i \in \{1, \ldots, K\}$ can be converted into a one-hot vector $\mathbf{y}_i$ with $y_{ik} = 1$ if the sample belongs to the k-th class and $y_{ij} = 0$ for $j \neq k$. By arranging $\boldsymbol{\alpha}_i$ and $\mathbf{y}_i$ into matrices $\mathcal{A} := [\boldsymbol{\alpha}_i]_{i \in \mathbb{L}}$ and $Y := [\mathbf{y}_i]_{i \in \mathbb{L}}$, the UCE loss function is defined as:

$$\mathrm{UCE}(\mathcal{A}, Y) = \sum_{i \in \mathbb{L}} \sum_{k \in [K]} y_{ik} (\Psi(\alpha_{i0}) - \Psi(\alpha_{ik})), \tag{14}$$

where $\alpha_{i0} = \sum_{k=1}^K \alpha_{ik}$ and Ψ is the digamma function given by

$$\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}.$$

B Detailed Framework of GPN

In this section, we review the architecture of GPN, followed by two examples that reveal a limitation of Theorem 1 in the GPN paper [32].

Multi-layer Perceptron (MLP). Instead of deep convolution layers used in many neural networks designed for image classification task [7], GPN utilizes two simple perceptron layers with ReLU activation function as the encoding network, which maps high dimensional data to a latent space with a much smaller dimension, avoiding the curse of dimensionality for the density estimation on a (mapped) latent representation [25]. As each node is independent of the others in this step, the encoding map only considers the node features without any graph structure involved. Mathematically, the mapping can be expressed by

$$\boldsymbol{z}_i = f(\boldsymbol{x}_i; \boldsymbol{\theta}) = W_2 \sigma(W_1 \boldsymbol{x}_i + \mathbf{1}^\intercal \mathbf{b}_1) + \mathbf{1}^\intercal \mathbf{b}_2,$$

where $\theta := \{W_1, W_2, \boldsymbol{b}_1, \boldsymbol{b}_2\}$ denotes a set of learning parameters. For simplicity we use the notation $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$.

Normalizing Flow. Normalizing flow is used to estimate the density $\mathbb{P}(\boldsymbol{z}_i | k; \boldsymbol{\phi})$ for $k \in [K]$ and learning parameters $\boldsymbol{\phi}$ as an invertible transformation $q(\cdot; k)$ of a base distribution, e.g. Normal distribution, which denotes the distribution of class k in the latent space. The default flow in GPN is the radial flow [27], given by

$$q(\mathbf{z}; k) = \mathbf{z} + \frac{\beta(\mathbf{z} - \mathbf{z}_0)}{\gamma + \|\mathbf{z} - \mathbf{z}_0\|}$$

$$\mathbb{P}(\mathbf{z}_i | k; \phi) = p_z(q^{-1}(\mathbf{z}_i; k)) |\det \frac{\partial q^{-1}(\cdot; k)}{\partial \mathbf{z}}|.$$

where $\mathbf{z}_0$ is a reference point and $p_z(\cdot) \sim \mathcal{N}(0, 1)$. After estimating the density of the node i belonging to a specific class k, the pseudo evidence counts are scaled to the probability, i.e., $\beta_i^k \propto \mathbb{P}(\boldsymbol{z}_i | k; \boldsymbol{\phi})$, GPN sets

$$\beta_i^k := g_{\boldsymbol{\phi}}(\boldsymbol{z}_i)_k = N_k \cdot \mathbb{P}(\boldsymbol{z}_i | k; \boldsymbol{\phi}),$$

where N_k is the number of training nodes that belong to the class k.

Personalized Page Rank. GPN applies a personalized page rank (PPR) module to diffuse the evidence among neighboring nodes. It is motivated by the work of Approximate Personalized Propagation of Neural Predictions (APPNP) [12] that is designed to decouple the prediction (only based on node features) with any encoding network and propagate with a personalized page rank (PPR) module (only based on edge information). In particular, PPR provides a personalized influence score matrix for each node that considers L hop of neighbors without involving any new parameters to learn and L is a hyperparameter:

$$\boldsymbol{\beta}^{(l+1)} = (1-\gamma)\hat{A}\boldsymbol{\beta}^{(l)} + \gamma\boldsymbol{\beta}^{(0)},$$

where γ is a hyper-parameter relating to the teleport probability, $\hat{A}$ denotes the symmetrically normalized graph adjacency matrix with added self-loops (i.e., $\hat{A} := D^{-1/2}AD^{-1/2}$ with the standard adjacency matrix A), and l denotes the layer index with $\boldsymbol{\beta}^{(0)}$ obtained after the normalizing flow. The output of PPR is a set of concentration parameters, denoted by $\boldsymbol{\alpha} = h_\gamma(\boldsymbol{\beta}^{(0)})$.

Collectively for MLP, normalizing flow, and PPR, the network in GPN can be expressed by

$$\boldsymbol{\alpha}_i = 1 + h_\gamma(g_\phi(f_\theta(\mathbf{x}_i))), \tag{15}$$

for each node i, where the addition of 1 guarantees that the concentration parameter is strictly positive. In addition, an entropy regularization was considered by GPN defined by,

$$\mathbb{H}(\mathrm{Dir}(\boldsymbol{\alpha}_i)) = \log B(\boldsymbol{\alpha}_i) + (\alpha_{i0} - K)\Psi(\alpha_{i0}) - \sum_{k=1}^{K}(\alpha_{ik} - 1)\Psi(\alpha_{ik}), \tag{16}$$

where $\alpha_{i0} = \sum_{k=1}^{K} \alpha_{ik}$.

Next, we provide two examples to describe oversight of [6, Theorem 1] and [32, Theorem 1] in the sense that both theorems assume an impossibility. Particularly the assumption is that a two-layer ReLU network can be represented by a set of affine mappings, each being full rank, from a finite set of regions to the latent space. However, we construct Example 9 and Example 10 to show this assumption is impossible.

Example 9. *We start with a simple case where a two-layer ReLU network with input, hidden layer, and output of a scalar (1-dimensional) is considered for an easier interpretation of the results. One simple example of a two-layer ReLU network is expressed by*

$$z = f_\theta(x) = 1 \cdot \sigma_{ReLU}(1 \cdot x + 0) + 0. \tag{17}$$

Following [15], we split the latent space into two affine regions, i.e.,

$$z = \begin{cases} x & \text{if } x \in [0,\infty) \\ 0 & \text{if } x \in (-\infty, 0], \end{cases} \tag{18}$$

labeled by $Q^{(0)} = [0,\infty)$ and $Q^{(1)} = (-\infty, 0]$. We see the associated $V^{(0)} = 1$ and $V^{(1)} = 0$ in the affine representation (17) that certainly do not have independent rows, as required by [6, Theorem 1].

Example 10 extends the 1D case in Example 9 into a higher d-dimension, showing that there is always at least one affine region that produces a single value, i.e. $f_\theta(Q^{(l)^*}) = \{\mathbf{v}\}$ when mapped into a ReLU network f_θ.

Example 10. *We consider the ReLU network,*

$$f_\theta(\mathbf{x}) = C\sigma_{ReLU}(B\mathbf{x}), \tag{19}$$

where $B, C \in \mathbb{R}^{d\times d}$ are matrices of full rank. Denote $\mathbf{x}_j$ to be the solution to the equation,

$$-\mathbf{e}_j = B\mathbf{x}, \tag{20}$$

where $\mathbf{e}_j$ is the jth Euclidean standard basis. As B is assumed to be full rank, there is the unique solution of the corresponding $\mathbf{x}_j$.

Notice that the polytope,

$$S = \left\{ \sum_{j=1}^{d} a_j \mathbf{x}_j \middle| a_j \geq 0 \right\}, \tag{21}$$

has non-zero measure in $\mathbb{R}^d$. Note that the ReLU network is constant by construction, as $\sigma_{ReLU}(-\mathbf{e}_j) = \mathbf{0}$. In other words, we have for $x \in S$ that

$$\mathbf{z} = f_\theta(\mathbf{x}) = C\sigma_{ReLU}(B\mathbf{x}) = C\mathbf{0} = \mathbf{0}. \tag{22}$$

As in the previous example, the existence of S means that there exists some $V^{(\cdot)} = \mathbb{0}_{d,d}$ which contradicts the assumption that all Vs have independent rows. Under this setting, the density does not approach zero, which is the conclusion of Theorem 1 in [32].

C Proofs

In this section, we provide the proof of all the theorems and corollaries. Note that until Theorem 8, We ignore the graph component h_γ and focus solely on the representational layer f_θ and normalizing flow layer g_ϕ.

Proof of Theorem 1 and Corollary 3. As f_θ is arbitrary by assumption, we choose it in such a way that it maps a point in $\mathcal{X}_k$ to a point inside the ball centered at $\mathbf{z}_k$ with radius r_k, denoted by $B(\mathbf{z}_k, r_k)$,

$$f_\theta : \mathcal{X}_k \to \mathcal{Z}_k \subset B(\mathbf{z}_k, r_k), \tag{23}$$

where $\mathbf{z}_k \in \mathcal{Z}$ with a minimal distance R between any two of them, i.e., $d(\mathbf{z}_k, \mathbf{z}_m) > R, \forall k, m \in [K]$, and we define $r > r_k, \forall k \in [K]$. We then choose the normalizing flow to be,

$$g_\phi(\mathbf{z}; k) = 1 + N_k \cdot \begin{cases} \frac{1}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))}, & \text{if } \mathbf{z} \in B(\mathbf{z}_k, r_k), \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

We add the value of 1 in the normalizing flow to produce valid evidence measures. We also assume that $N_k = \mu(\mathcal{X}_k) > 0$, where μ is the Lebesgue measure function.

The global minimum of UCE occurs when UCE is equal to 0 for every class. Recall that

$$\sum_{k \in [K]} \mathrm{UCE}\left(g(\mathcal{Z}_k), Y\right) = \sum_{k \in [K]} \int_{\mathcal{Z}_k} \left(\Psi\left(\sum_{m \in [K]} g_m(\mathbf{z}) \right) - \Psi(g_k(\mathbf{z})) \right) d\mu. \tag{25}$$

Using (23), we consider an upper bound of the right-hand side by integrating over the larger region, that is,

$$\sum_{k \in [K]} \int_{B(\mathbf{z}_k, r_k)} \left(\Psi\left(\sum_{m \in [K]} g_m(\mathbf{z}) \right) - \Psi(g_k(\mathbf{z})) \right) d\mu, \tag{26}$$

$$= \sum_{k \in [K]} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) \cdot \left(\Psi\left(K + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right) - \Psi\left(1 + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right) \right). \tag{27}$$

According the recurrence relation of the digamma function: $\Psi(x+1) = \Psi(x) + 1/x$, we readily derive that,

$$\sum_{k \in [K]} \mathrm{UCE}\left(g(\mathcal{Z}_k), Y\right) \leq \sum_{k \in [K]} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) \cdot \sum_{m=1}^{K-1} \left(K - m + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right)^{-1}. \tag{28}$$

Taking the limit of the right-hand side yields

$$\lim_{r \to 0} \sum_{k \in [K]} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) \cdot \sum_{m=1}^{K-1} \left(K - m + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right)^{-1}, \tag{29}$$

$$= \sum_{k \in [K]} \lim_{r_k \to 0} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) \cdot \lim_{r_k \to 0} \sum_{m=1}^{K-1} \left(K - m + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right)^{-1}. \tag{30}$$

It is straightforward for the following two limits to hold,

$$\lim_{r_k \to 0} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) = 0, \tag{31}$$

$$\lim_{r_k \to 0} \sum_{m=1}^{K-1} \left(K - m + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right)^{-1} = 0, \tag{32}$$

thus leading to

$$\lim_{r \to 0} \sum_{k \in [K]} \mathrm{Vol}(B(\mathbf{z}_k, r_k)) \cdot \sum_{m=1}^{K-1} \left(K - m + \frac{N_k}{\mathrm{Vol}(B(\mathbf{z}_k, r_k))} \right)^{-1} = 0. \tag{33}$$

On the other hand, as $\Psi(\sum_{k \in [K]} g_k(\mathbf{z})) - \Psi(g_k(\mathbf{z})) \geq 0$, we have

$$0 \leq \sum_{k \in [K]} \int_{\mathcal{Z}_k} (\Psi \left(\sum_{m \in [K]} g_m(\mathbf{z}) \right) - \Psi(g_k(\mathbf{z})) d\mu = \sum_{k \in [K]} \mathrm{UCE}\left(g(\mathcal{Z}_k), Y\right), \tag{34}$$

which implies that UCE $\to 0$ as $r \to 0$. For $r = 0$, UCE is equal to zero, which leads to Corollary 3. □

Proof of Theorem 4. We denote the parameters of g_ϕ that represent the true analytic solution $\hat{\phi} = \phi(\theta)$. Note that this is a function with respect to the choice of θ, that is, the true distribution is dependent on the representational mapping. In this proof, we focus on finding a value of θ s.t.,

$$\hat{\theta} = \arg\min_{\theta} \text{UCE}(\alpha(\theta, \hat{\phi}), Y) = \arg\min_{\theta} \text{UCE}(\alpha(\theta, \phi(\theta)), Y). \tag{35}$$

As the true distribution is dependent on the representational mapping, we should consider a joint minimization problem with respect to both the representation map and density distribution.

We will separate the proof into two cases. Specifically, we prove Case 1 by contradiction, showing that if the set $\mathcal{Z}_k$ mapped to by f_θ from $\mathcal{X}_k$ has a non-zero measure, then the global minimizer UCE $= 0$ can not be achieved. We then prove Case 2, under the assumption that a true analytical solution may achieve density evidence at a point, by showing that we may achieve the global minimizer on a point set.

Case 1: Non-Zero Measure. Suppose the true distribution on this set is a non-degenerate distribution. As the natural definitions of a probability distribution $1 = \int_{\mathcal{Z}_k} d\mu$, the UCE loss can be expressed by

$$\sum_{k\in[K]} \text{UCE}\left(g(\mathcal{Z}_k), Y\right) = \sum_{k\in[K]} \int_{\mathcal{Z}_k} \left(\Psi\left(\sum_{m\in[K]} g_m(\mathbf{z}) \right) - \Psi(g_k(\mathbf{z})) \right) d\mu(\mathbf{z}). \tag{36}$$

In order for the measure of $\mathcal{Z}_k$ to have a density of $\epsilon > 0$, there exists a subset of $\mathcal{Z}_k$ with non-zero measure $\delta_k > 0$, denoted $\mathcal{Z}_k^*$. Using similar techniques as the proof of Theorem 1 in reverse, we obtain,

$$\sum_{k\in[K]} \text{UCE}\left(g(\mathcal{Z}_k), Y\right) \geq \sum_{k\in[K]} \int_{\mathcal{Z}_k} \left(\Psi\left(K + N_k\epsilon\right) - \Psi(1 + N_k\epsilon)\right) d\mu,$$

then for some $k \in [K]$ there exists some $\mathcal{Z}_k^*$,

$$\begin{aligned} \sum_{k\in[K]} \text{UCE}\left(g(\mathcal{Z}_k), Y\right) &\geq \int_{\mathcal{Z}_k^*} \left(\Psi\left(K + N_k\epsilon\right) - \Psi(1 + N_k\epsilon)\right) d\mu, \\ &= \delta_1 \cdot \left(\Psi\left(K + N_k\epsilon\right) - \Psi(1 + N_k\epsilon)\right). \end{aligned}$$

As Ψ is strictly increasing, then $\delta_2 = \Psi\left(K + N_k\epsilon\right) - \Psi(1 + N_k\epsilon) > 0$, which implies that

$$\sum_{k\in[K]} \text{UCE}\left(g(\mathcal{Z}_k), Y\right) \geq \sum_{k\in[K]} \delta_1 \cdot \delta_2 > 0.$$

Therefore, we prove that if f_θ maps to a measurable set, the UCE loss is necessarily non-zero.

Case 2: Zero Measure Sets. Corollary 3 shows that the zero UCE is achievable. If Case 1 fails, then we can conclude that only on a disjoint set $\mathcal{Z}_k$ with measure 0 for each k is permissible to achieve the UCE to be 0. The exact choice of this set depends on the precise definitions of the probability distributions on a point set and their ability to achieve infinite densities. Here we constrain these possibilities by requiring the range of f_θ to have non-zero measure or to be a point set if having zero measure[2]. □

Proof of Theorem 6. Pick $\mathbf{x} \in \mathcal{X}$ s.t. $d(\mathbf{x}, \mathbf{x}_k) > \delta$ for $x_k \in \mathcal{X}_k$ see that for any f_θ where $\theta \in \Gamma$ we have that $\mathbf{x}$ is necessarily not mapped to $\mathbf{z}_k \in \mathcal{Z}$ (if it were mapped in $\mathcal{Z}$ then the preimage would contain it and thus we would have $d(\mathbf{x}, \mathbf{x}_k) < \delta$, which is a contradiction with our selection of x). Recall that the density for any point mapped to z_k is infinite. That is the density of the associated points mapped to the point set is necessarily infinite and the density of points mapped elsewhere is necessarily smaller, namely 0, with the associated evidence 1. Our selected $\mathbf{x}$ then has no evidence in favor of it belonging to a class k while any point in $\mathbf{x}_k \in \mathcal{X}_k$ must have infinite evidence by our choice of a well-fit θ. □

Proof of Corollary 7. Notice that we can choose two types of such θ,

Case 1 Let θ for class k be chosen such that,

$$f(\mathbf{x}) = \begin{cases} \mathbf{z}_k \text{ if } \mathbf{x} \in \mathcal{X}_k, \\ 0 \text{ otherwise.} \end{cases} \tag{37}$$

[2] We choose that the cardinality of the zero-measure set $f_\theta(\mathcal{X}_k)$ to be finite (rather than countably infinite) as we do not want to detail precise topological arguments (like compactness and boundedness) about the pointsets and their respective preimages.

Case 2

$$f(x) = \begin{cases} \mathbf{z}_k \text{ if } d(\mathbf{x}, \hat{\mathbf{x}}) < \delta \text{ for any } \hat{x} \in \mathcal{X}_k, \\ 0 \text{ otherwise.} \end{cases} \tag{38}$$

If $\mathbf{x} \in \mathcal{X}_k$ is mapped to $\mathbf{z}_k$ then it is endowed with infinite density, moreover, it is believed to be an ID node belonging to class k. Thus, the nearby OOD being detected for these UCE minimizers is determined by arbitrary choice. □

Proof of Theorem 8. First note that the ID nodes are mapped to have infinite evidence achieved at the points in the latent space $\mathcal{Z}_k$. As the representations of the OOD nodes are in $\mathcal{Z}_k$ they are also endowed with infinite evidence. That is graph layers can only help separate nodes by pulling them towards the center of their own classes w.r.t. to the representation space this is only helpful if their representations are separate to begin with.

□

Lastly, we give a toy example showing heuristically that the proposed regularization yields a better separation of the OOD nodes from IDs, compared to the original GPN model without the distance-based regularization.

Example 11. *Consider two ID classes (Class 1 and Class 2) and one OOD class with the following construction:*

1. *All nodes belonging to Class* 1 *have feature values sampled from* $\mathbf{x}^{(1)} = [1, 0, 0]$.
2. *All nodes belonging to Class* 2 *have feature values sampled from* $\mathbf{x}^{(2)} = [-1, 0, 0]$.
3. *All nodes belonging to the OOD class* 2 *have feature values sampled from* $\mathbf{x}^{(OOD)} = [0, 1, v]$.
4. *We sample v from the uniform distribution $U(-1, 1)$ independently for each sample in each class.*
5. *All nodes are connected to every node within their own class, leading to a graph of homophily 1.*
6. *Suppose the density function is true density distribution*
7. *Denote the PPR layer by $\hat{h}$ that uses the right normalized adjacency matrix AD^{-1} rather than symmetrically normalized $D^{-1/2}AD^{-1/2}$, used in APPNP.*
8. *Suppose f_θ is a linear function (i.e. no activation function) explicitly, $W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{bmatrix}$.*

Then GPN with our regularization can learn an embedding that makes it possible to separate classes 1, 2, and OOD nodes. Without regularization, OOD nodes lie between ID classes in the latent space.

Proof. A simple calculation for the project leads to

$$\mathbf{z}_i = [XW]_i = \begin{cases} [W_{11}, W_{12}] & \text{for class 1 nodes} \\ [-W_{11}, -W_{12}] & \text{for class 2 nodes} \\ [W_{21} + vW_{31}, W_{22} + vW_{31}] & \text{for OOD nodes.} \end{cases} \tag{39}$$

Clearly, the values of W_{31} and W_{32} would be smaller with the distance minimization term applied than without, as v is selected randomly. Moreover neither W_{31} nor W_{32} affects the model's ability to separate the two classes as desired. We explicitly calculate both UCE and the distance-based regularization in the objective function, while ignoring the Dirichlet regularization, thus leading to the following objective function,

$$L(Z, \alpha, Y; \mathcal{G}) = \text{UCE}(\hat{h}(g_\phi(f_\theta(x))), Y) + R(Z; \mathcal{G}). \tag{40}$$

First, we explicitly work out the distance-based regularization term

$$\begin{aligned} \mathrm{R}(Z; \mathcal{G}) &= \sum_{(i,j)\in\mathcal{E}} \|\mathbf{z}_i - \mathbf{z}_j\|^2 \\ &= \sum_{(i,j)\in\mathcal{E}_1} \|[W_{11}, W_{12}] - [W_{11}, W_{12}]\|^2 + \sum_{(i,j)\in\mathcal{E}_2} \|[-W_{11}, -W_{12}] - [-W_{1,1}, -W_{12}]\|^2 \\ &\quad + \sum_{(i,j)\in\mathcal{E}_{\text{OOD}}} \left\| [W_{21} + v^{(i)}W_{31}, W_{22} + v^{(i)}W_{32}] - [W_{21} + v^{(j)}W_{31}, W_{22} + v^{(j)}W_{32}] \right\|^2 \\ &= \sum_{(i,j)\in\mathcal{E}_{\text{OOD}}} \left\| (v^{(i)} - v^{(j)})[W_{31}, W_{32}] \right\|^2, \end{aligned}$$

which is minimized when W_{31}, W_{32} go to zero.

Next, we consider the UCE loss portion. See that as we estimate the true density using g we will have no overlap between the two distributions $\mathcal{Z}_1, \mathcal{Z}_2$. We are left with $W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ 0 & 0 \end{bmatrix}$,

$$Z_i = [XW]_i = \begin{cases} [W_{11}, W_{12}] \text{ for class 1 nodes} \\ [-W_{11}, -W_{12}] \text{ for class 2 nodes} \\ [W_{21}, W_{22}] \text{ for OOD nodes,} \end{cases} \tag{41}$$

If W is to remain full rank this will necessarily require either W_{21} or W_{22} to be non-zero. Thus OOD nodes will be mapped as we see in (41) to some distinct values - which can be separated after the application of APPNP as we expect APPNP to only average the values within each class. □

D Additional Experimental Details

D.1 Descriptions of Baselines

Graph-based Kernel Dirichlet distribution Estimation (GKDE) [39]: Based on the high homophily property of most graphs (neighboring nodes tend to share the same class label), GKDE derives the evidence with the help of the node-level distances (shortest path in the graph) with training nodes belonging to the same class.

Label Propagation (LP) [32]: Following the idea of GKDE, LP collects the evidence by relying on the density of labeled nodes in neighborhoods rather than distance. An initial condition per class is defined and then a Personalized Page Rank is used as the diffusion.

VGCN-Energy [21]: It is a GCN-based model with energy score as the uncertainty estimation which maps each node to a single, non-probabilistic scalar called the energy. The energy score can be calculated as follows

$$s^i_{\text{energy}} = -T \log \sum_{k=1}^{K} \exp^{\frac{\boldsymbol{l}_i^k}{T}},$$

where $\boldsymbol{l}$ is the predicted logits of a neural network and temperature parameter $T = 1$.

GKDE-GCN [39]: GKDE-GCN utilizes a GCN network to estimate the multisource uncertainty by a Dirichlet distribution and then sample probability as well as the class prediction. The evidence derived from the aforementioned GKDE is as a teacher of concentration parameters of Dirichlet Distribution, and another deterministic GCN predicting the probability is used as a teacher for sampled probability. The overall loss is composed of the KL divergence between these two teachers with the corresponding distribution and Bayes risk with respect to the squared loss of sampled class prediction.

APPNP [12]: Given that message passing neural network suffers from the over-smoothing problem that limits the depth of the neural network, APPNP proposed to decouple the prediction and propagation where the prediction depends on the node features and propagation depends on interactions between nodes through edges. APPNP first uses any kind of neural network to embed the input space and diffuses information with a personalized page rank. For large graphs, they use power iteration to approximate a topic-sensitive page rank.

GPN [32]: GPN applies a normalizing flow to estimate the density of each class in the latent space embedded with an encoding network and then propagates the scaled density as the evidence.

D.2 Description of Datasets

We use three citation networks, labelled by CoraML, CiteSeer, Pubmed [4], two co-purchase Amazon datasets [31], labeled by Computers and Photos, two coauthor datasets [31], labeled by CoauthorCS and Physics, and a large dataset OGBN Arxiv [16]. We use the same train/val/test split of 5/15/80 as [32]. The details of the graphs and setups for the OOD detection are provided in Table 4.

Table 4: Dataset Description

	CoraML	CiteSeer	PubMed	Computers	Photos	Coauthor CS	Coauthor Physics	OGBN-Arxiv
#nodes	2,995	4,230	19,717	13,752	7,650	18,333	34,493	169,343
#edges	16,316	10,674	88,648	491,722	238,162	163,788	495,924	2,315,598
#features	2879	602	500	767	745	6,805	8,415	128
#classes	7	6	3	10	8	15	5	40
# left-out-classes	3	2	1	5	3	4	2	15

D.3 Hyper-parameter tuning

We follow the same setting with [32]. In detail, we use the Adam optimizer with a learning rate of 0.01. For VGCN-Energy, we use a temperature of $T = 1.0$. We carefully tune three hyperparameters: the distance-based regularization weight, Dirichlet entropy weight, and activation functions. We select the best parameters for each dataset separately that returns the highest validation cross-entropy. The detailed hyperparameters configuration is as Table 5.

Table 5: Hyperparameter configurations of proposed model

	Dirichlet Entropy Reg. Weight	Graph Distance Reg. Weight	Activation function
CoraML	0	10^{-4}	GELU
CiteSeer	10^{-4}	$10^{-9.5}$	LogSigmoid
PubMed	10^{-5}	10^{-4}	RELU
Computers	10^{-5}	10^{-4}	RELU
Photos	10^{-5}	10^{-11}	RELU
Coauthor CS	0	10^{-6}	RELU
Coauthor Physics	10^{-4}	$10^{-4.5}$	LogSigmoid
OGBN-Arxiv	10^{-5}	10^{-8}	RELU

We also consider the following activation functions in the encoding network with element-wise operations,

$$\begin{aligned}
\sigma_{\text{RELU}}(x) &= \max(0, x), \\
\sigma_{\text{LogSigmoid}}(x) &= \log\left((1+\exp(-x))^{-1}\right), \\
\sigma_{\text{GeLU}}(x) &= x\text{CDF}_{\mathcal{N}}(x) \\
\sigma_{\text{HardTanh}}(x) &= \begin{cases} -1, x < -1 \\ x, -1 \leq x \leq 1 \\ 1, x > 1 \end{cases}.
\end{aligned}$$

ReLU is the most popular activation function used in the hidden layer of neural networks, which brings efficient computation by only activating neurons with positive outputs. Sigmoid is popularly used for probability prediction because its output is always in the range (0,1) with a smooth gradient. GeLU has better nonlinearity and is widely used in Natural Language processing and computer vision. HardTanh is a more computation-efficient version of Tanh.

E Additional Experiments

E.1 Additional Experiments - OOD Detection

For Amazon Photos, Amazon Computers, Coauthor CS, Coauthor Physics, and OGBN Arxiv dataset, the OOD Detection results are shown in Table 6.

E.2 Additional Experiments - Misclassification Detection

For Amazon Photos, Amazon Computers, Coauthor CS, Coauthor Physics, and OGBN Arxiv dataset, the Misclassification Detection results are shown in Table 7.

E.3 Graph Distance Minimization

We plot the tSNE visualization of latent space with different distance-based regularization weights and symbol sizes denote the total evidence. We plot for coraML in Figure 2, CiteSeer in Figure 3, Coauthor CS in Figure 4, Coauthor Physics in Figure 5. With increasing weight, it tends to have a more separable latent representation for different categories while degenerate mappings occur when distance minimization is too large.

Table 6: OOD Detection (Cont.)

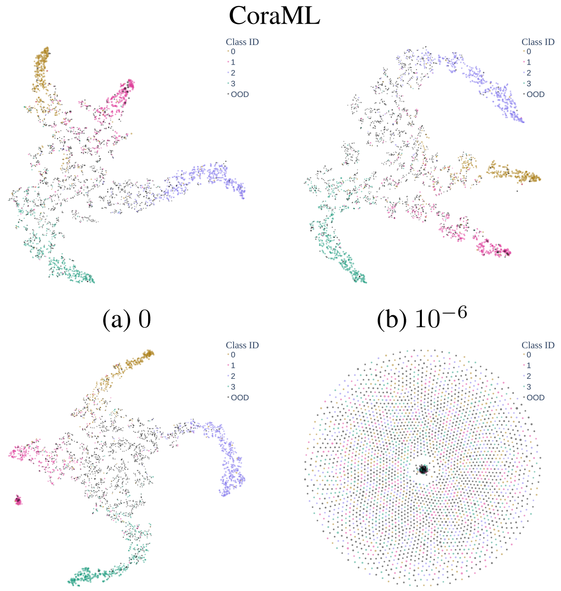
Data	Model	ID-ACC	AUROC			AUPR		
			Alea w/	Epi w/	Epi w/o	Alea w/	Epi w/	Epi w/o
Amazon Computers	LP	83.28	**86.74**	83.88	n.a.	**67.10**	63.08	n.a.
	GKDE	71.41	75.14	73.58	n.a.	49.21	47.68	n.a.
	VGCN-Energy	88.95	82.76	83.43	n.a.	57.49	60.64	n.a.
	GKDE-GCN	82.73	77.03	70.32	n.a	49.81	45.92	n.a
	GPN	88.48	82.49	87.63	**74.55**	56.78	67.94	**48.03**
	Ours	**89.88**	83.56	**89.26**	71.82	58.51	**71.06**	43.35
Amazon Photos	LP	89.27	**94.24**	90.26	n.a.	**90.24**	85.55	n.a.
	GKDE	85.94	76.51	60.83	n.a.	66.72	59.09	n.a.
	VGCN-Energy	94.24	82.44	79.64	n.a.	72.60	71.71	n.a.
	GKDE-GCN	89.84	73.65	69.09	n.a	62.45	59.68	n.a
	GPN	94.10	82.72	91.98	76.57	74.55	86.29	64.00
	Ours	**94.40**	83.51	**92.30**	**78.10**	77.65	**87.36**	**65.39**
Coauthor CS	LP	86.40	83.78	80.86	n.a.	74.8	71.15	n.a
	GKDE	78.84	79.32	77.59	n.a.	66.30	64.69	n.a.
	VGCN-Energy	93.07	**85.35**	87.33	n.a.	**80.87**	82.79	n.a.
	GKDE-GCN	**93.13**	85.02	84.45	n.a.	80.15	77.90	n.a.
	GPN	88.21	69.49	**92.90**	88.84	55.41	90.28	86.54
	Ours	89.24	70.12	92.37	**91.38**	56.20	**91.17**	**90.45**
Coauthor Physics	LP	95.39	**91.78**	90.03	n.a.	**70.58**	69.63	n.a.
	GKDE	93.30	87.02	84.64	n.a.	57.00	52.49	n.a.
	VGCN-Energy	**97.96**	90.29	91.08	n.a.	63.63	69.41	n.a.
	GKDE-GCN	97.95	87.38	84.62	n.a.	57.97	56.30	n.a.
	GPN	97.40	85.20	**94.51**	89.63	61.89	**83.73**	66.44
	Ours	97.44	85.28	94.42	**90.36**	62.80	83.61	**70.62**
OGBN Arxiv	LP	66.84	**80.04**	**75.22**	n.a.	**65.21**	**67.69**	n.a.
	GKDE	51.51	68.12	65.80	n.a.	47.22	45.23	n.a.
	VGCN-Energy	**75.61**	64.91	64.50	n.a.	42.72	42.41	n.a
	GKDE-GCN	73.89	68.84	72.44	n.a.	49.71	52.23	n.a.
	GPN	73.84	66.33	74.82	62.17	46.35	58.71	**43.01**
	Ours	71.30	66.98	74.52	**62.75**	47.48	56.97	41.48

Alea: Aleatoric, Epi.: Epistemic, w/: with propagation, w/o: without propagation

Table 7: AUROC and AUPR for the Misclassification Detection (Cont.)

Data	Model	AUROC		AUPR	
		Alea w/	Epi w/	Alea w/	Epi w/
Amazon Computers	APPNP	79.75	n.a.	45.10	n.a.
	VGCN-Energy	82.08	n.a.	45.53	n.a.
	GKDE-GCN	79.66	73.66	63.26	56.93
	GPN	**82.20**	**77.58**	47.93	41.80
	Ours	80.75	74.87	**93.12**	**90.11**
Amazon Photos	APPNP	85.74	n.a.	37.00	n.a.
	VGCN-Energy	**87.94**	n.a.	48.35	n.a.
	GKDE-GCN	84.11	75.07	54.35	45.43
	GPN	87.21	**83.38**	46.32	37.07
	Ours	84.42	81.61	**96.89**	**96.70**
Coauthor CS	APPNP	**89.92**	n.a.	37.98	n.a.
	VGCN-Energy	89.46	n.a.	38.86	n.a.
	GKDE-GCN	89.24	80.98	39.30	30.52
	GPN	85.72	81.56	46.12	38.98
	Ours	86.21	**83.94**	**97.34**	**96.80**
Coauthor Physics	APPNP	**93.27**	n.a.	38.14	n.a.
	VGCN-Energy	92.86	n.a.	37.19	n.a.
	GKDE-GCN	92.77	86.12	37.08	25.13
	GPN	91.14	**89.63**	41.43	35.64
	Ours	89.93	88.83	**99.14**	**99.10**
OGBN Arxiv	APPNP	77.55	n.a.	54.57	n.a.
	VGCN-Energy	**77.89**	n.a.	54.87	n.a.
	GKDE-GCN	77.47	**77.55**	61.62	62.33
	GPN	75.44	72.71	55.64	52.99
	Ours	75.30	72.85	**83.95**	**81.54**

Alea: Aleatoric, Epi.: Epistemic, w/: with propagation


CoraML

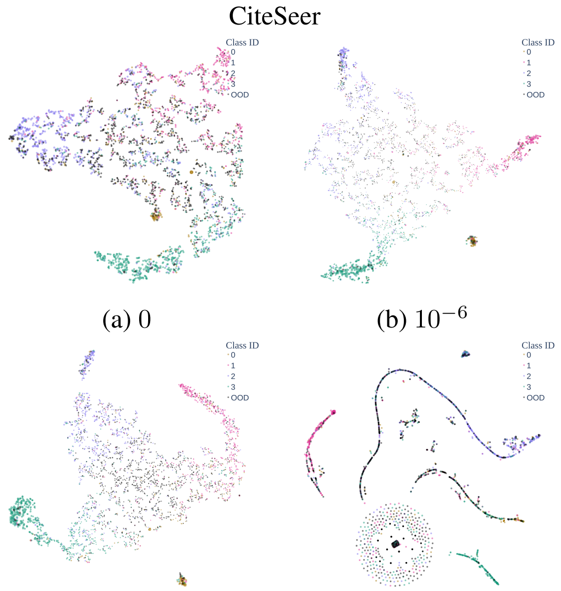
(a) 0

(b) 10^{-6}

(c) 10^{-4}

(d) 10^{-2}

Figure 2: latent representation for CoraML


CiteSeer

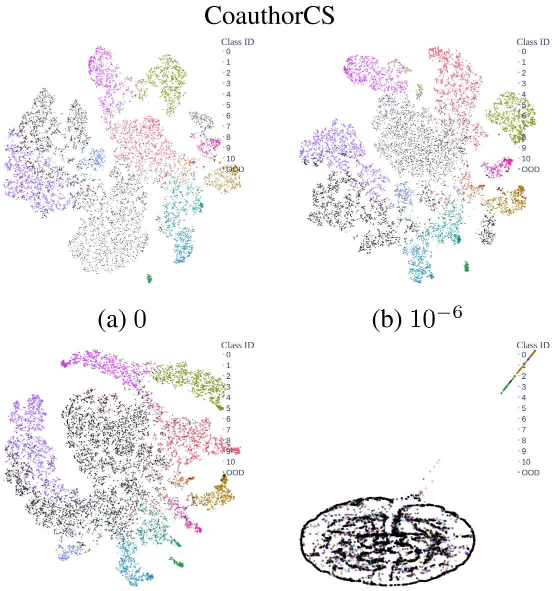
(a) 0

(b) 10^{-6}

(c) 10^{-4}

(d) 10^{-2}

Figure 3: latent representation for CiteSeer



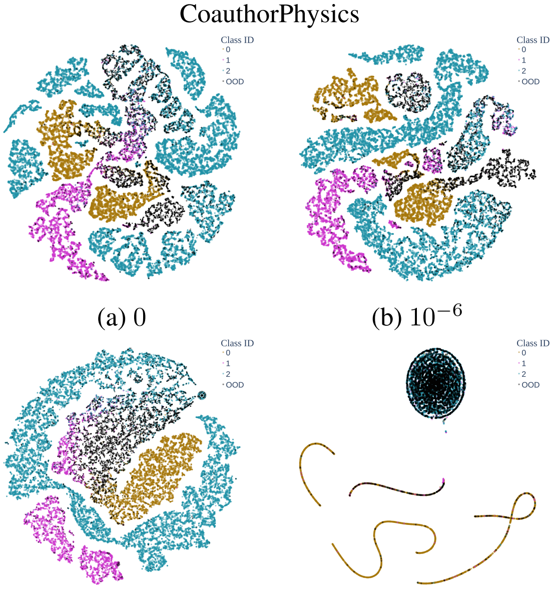
(a) 0

(b) 10^{-6}

(c) 10^{-4}

(d) 10^{-2}

Figure 4: latent representation for Coauthor CS



(a) 0

(b) 10^{-6}

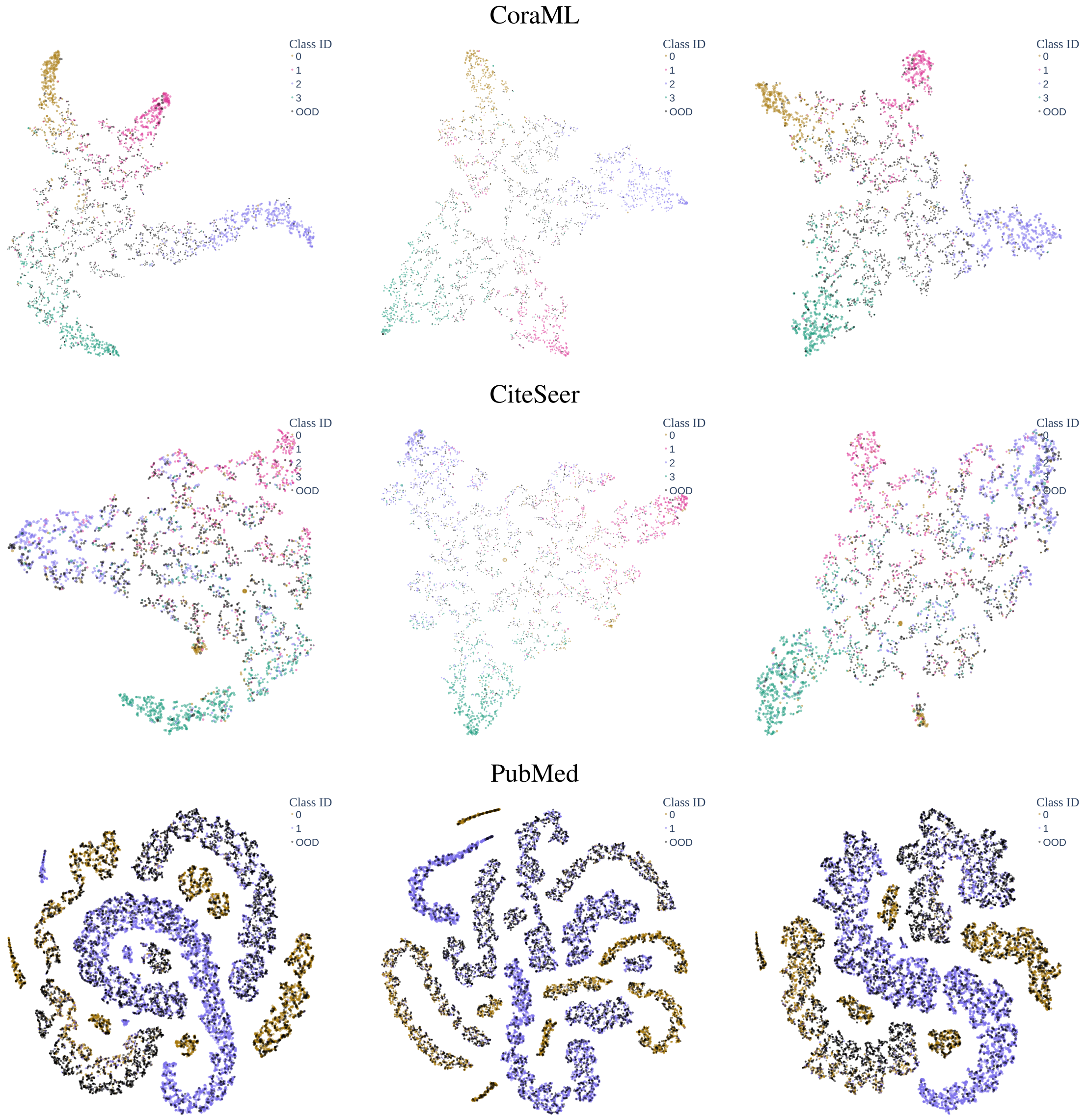
(c) 10^{-4}

(d) 10^{-2}

Figure 5: latent representation for Coauthor Physics

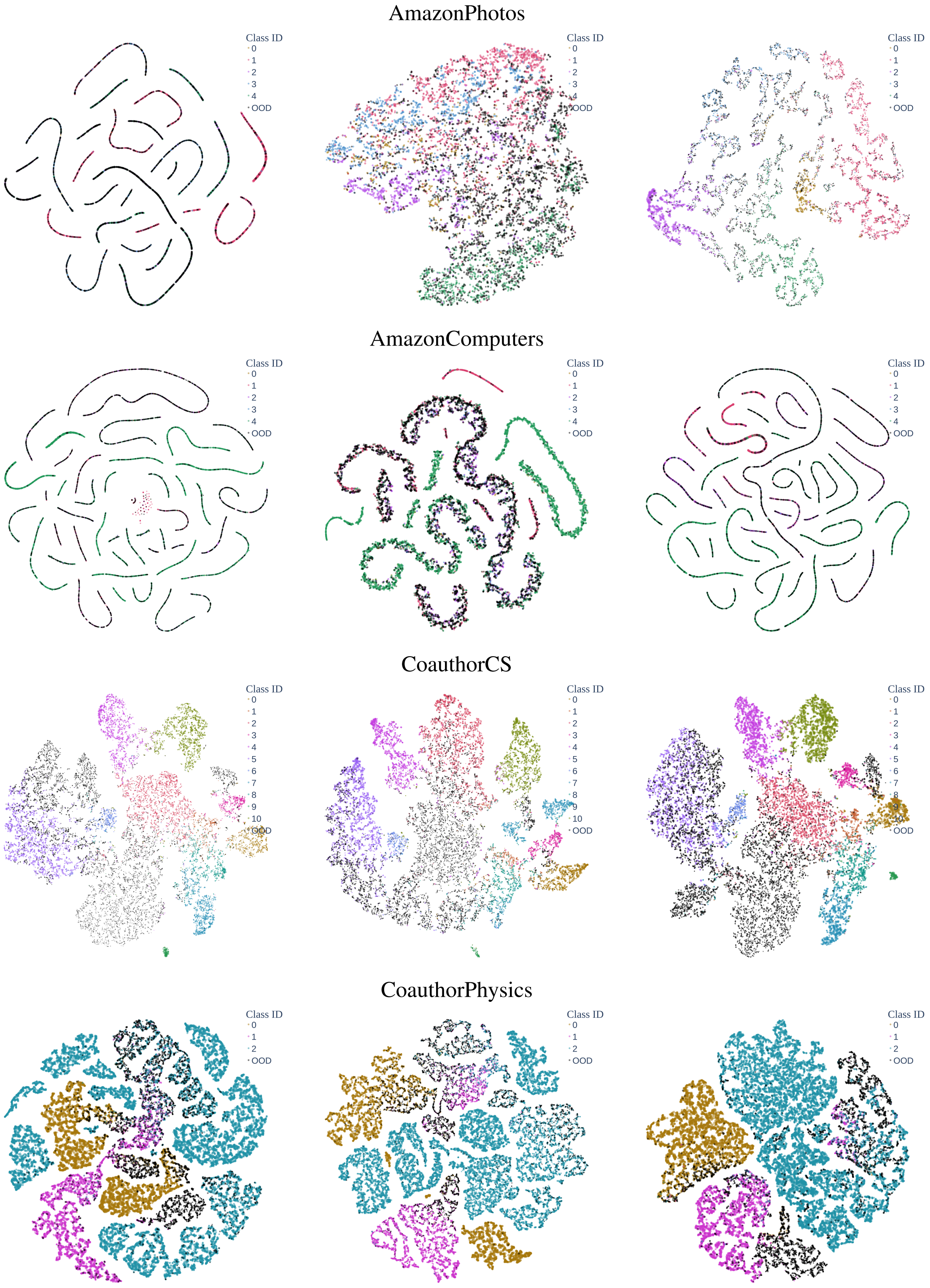
E.4 Graph Activation

In this subsection, we present the t-SNE visualizations of the learned representational space for various datasets in the following figures, without applying distance regularization. Instead, we introduce different activation functions. It is worth noting the notable distinction in quality when using the LogSigmoid activation function, which appears to be the smoothest among the activation functions employed on CiteSeer and Amazon Computers datasets. Once again, the size of each node corresponds to the square root of the learned evidence. Additionally, the color black indicates out-of-distribution (OOD) instances across all datasets, while distinct colors represent different classes.



(a) RELU (b) LogSigmoid (c) HardTanh

Figure 6: Latent representation for CoraML, CiteSeer and PubMed on different graph activation functions: RELU, LogSigmoid, and HardTanh.



(a) RELU

(b) LogSigmoid

(c) HardTanh

Figure 7: Latent representation for AmazonPhotos, AmazonComputers, CoauthorCS and CoauthorPhysics on different graph activation functions: RELU, LogSigmoid, and HardTanh.

E.5 Ablation Study

We show the full ablation study on three datasets: CoraML, CiteSeer and PubMed in Table 8.

Table 8: Ablation Study with OOD Detection task (cont.)

Data	Model	ID-ACC	AUROC			AUPR		
			Alea w/	Epi w/	Epi w/o	Alea w/	Epi w/	Epi w/o
CoraML	GPN	88.51	83.25	86.28	**80.95**	75.79	79.97	72.81
	GPN-CE	89.31	82.58	83.91	80.88	**76.54**	77.60	**76.05**
	GPN-CE-ACT	89.87	83.34	86.96	75.60	74.96	79.74	62.73
	GPN-CE-ACT-GD	**90.06**	**83.94**	**87.20**	76.12	76.26	**80.36**	63.32
Citeseer	GPN	69.79	72.46	70.74	66.65	55.14	50.52	44.93
	GPN-CE	70.98	74.20	73.75	68.41	58.12	53.55	46.60
	GPN-CE-ACT	71.96	74.72	77.97	72.28	60.41	56.04	50.73
	GPN-CE-ACT-GD	**72.51**	**75.22**	**78.98**	**73.21**	**62.30**	**58.63**	**52.73**
PubMed	GPN	**94.08**	71.84	73.91	71.2	57.92	67.19	59.72
	GPN-CE	93.84	74.19	78.32	74.50	59.85	74.11	64.55
	GPN-CE-ACT	93.84	74.19	78.32	74.50	59.85	74.11	64.55
	GPN-CE-ACT-GD	93.84	**75.23**	**81.76**	**77.79**	**60.75**	**78.16**	**69.19**

[*] Alea: Aleatoric, Epi.: Epistemic, w/: with propagation

GPN is the original results from the GPN paper with default hyperparameters and ReLU as the middle activation function, GPN-CE is the original GPN model with re-tuned dirichlet entropy regularization weight; GPN-CE-ACT is the original GPN model with re-tuned entropy regularization weight and activation function; GPN-CE-ACT-GD/(Ours) add the distance-based regularization term and tuned the two weights and activation function.