# Deep Hierarchical Knowledge Loss for Fault Intensity Diagnosis

## A Preliminaries

### A.1 Definition of Cavitation Intensity

Figure A1 demonstrates the variation in local pressure in a one-dimensional flow. Cavitation does not occur and the valve functions normally when the minimum pressure $p_{min}$ exceeds the vapor pressure $p_v$. However, when $p_v > p_{min}$, cavitation begins. In practice, direct measurement is difficult because the minimum pressure is downstream of the restriction. Therefore, the cavitation coefficient $X_{FZ}$ is proposed, which represents the ratio of the external pressure difference to the internal pressure difference. This coefficient can be determined empirically by assuming that cavitation noise begins when the minimum pressure $p_{min}$ matches the vapor pressure $p_v$. Consequently, the cavitation coefficient $X_{FZ}$ can be measured through the noise, which varies depending on the valve load. The formulas for the cavitation coefficient $X_{FZ}$ and the operating pressure ratio $X_F$ are given below:

$$X_{FZ} = \frac{p_u - p_d}{p_u - p_{min}}, \qquad X_F = \frac{p_u - p_d}{p_u - p_v}, \tag{A.1}$$

where $p_u$ represents the upstream pressure, $p_d$ denotes the downstream pressure, $p_{min}$ refers to the minimum pressure within the valve and $p_v$ stands for the vapor pressure.

When all coefficients are determined across the entire range of valve opening, the following conclusions can be drawn:

- $X_F < X_{FZ}$: The valve operates without cavitation and the flow may be either turbulent or laminar.
- $X_F \geq X_{FZ}$: When $X_F = X_{FZ}$, the valve operates exhibits incipient cavitation. As the difference between $X_{FZ}$ and $X_F$ increases, the cavitation region expands due to pressure drop from higher flow velocities.
- $X_F > 1$: Here the bubbles do not implode in the valve but rather continue to flow into the pipe because the downstream pressure $p_d$ is lower than the vapor pressure $p_v$. This phenomenon is called flashing.

The cavitation coefficient $X_{FZ}$ is applied only to the fluid, where it is measured empirically. Its valve varies for different liquid mediums due to changes in viscosity, dissolved gas content and other factors.

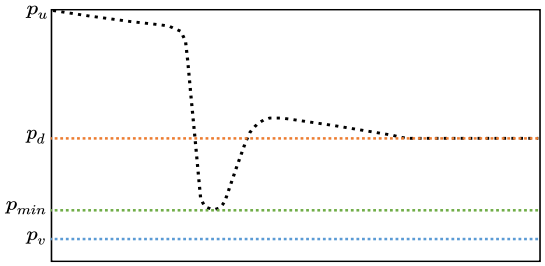



**Figure A1: Pressure flow model in a valve.**

### A.2 Cavitation Event Intensity Diagnosis

Cavitation is defined as the phenomenon of involving the formation, growth and collapse of local bubbles or vapor cavities in a liquid. In piping systems, the acoustic signals of different flow conditions (different levels of cavitation or non-cavitation) are recorded as continuous waveforms using acoustic sensors. Each observed acoustic signal records the entire physical process from the beginning to the end of the event of the corresponding flow state. In our experiments, cavitation intensity recognition mainly distinguishes incipient cavitation, constant cavitation, choked flow cavitation and non-cavitation. Whether severe or subtle, any form of cavitation would indicate a potential issue or failure in system operation. Therefore, it is crucial for operators of industrial systems to effectively and precisely recognize different intensities of cavitation in order to implement appropriate countermeasures.

### A.3 Acoustic Signals Augmentation

Formally, there is $x \subseteq \mathbb{R}^{M \times N}$ with $M$ measurements for each acoustic signal. Considering the purposely maintained steady flow status (i.e., it's always the same fluid status class within the individual measurement duration with 3 s or 25 s) in each recorded stream and fine resolution for the sensor. Each signal can split each stream into several pieces, each containing sufficient information for detection. The segment length is not too short to account for the inherent randomness in the noise emission and the features of each segment are independent. Therefore, we apply a sliding window (SW) with window size $w$ and step size $s$ to divide the acoustic signal sequence into a set of sub-sequences $\mathbf{X}_{sw} = \{x_{i,j}, i = 1, 2, \ldots, N; j = 1, 2, \ldots, k\} \subseteq \mathbb{R}^{w \times kN}$, where $k = \frac{(M-w)}{s}$ is the number of sub-sequences and $N$ is stream. The SW technique is a crucial part of the acoustic signal pre-processing.

### A.4 Time-Frequency Transform

Time-Frequency (T-F) transform provides more detailed and comprehensive information across both time and frequency dimensions. The most widely used T-F transform is computed by the short time Fourier transform (STFT), which can be converted back to time-domain signals by the inverse STFT (iSTFT). Given a sequence of signal $x[n]_{n=0}^{N-1}$, the STFT converts the signal sequence into the T-F domain and is defined by the formula:

$$\tilde{X}_w[k] = \sum_{n=0}^{N-1} x_w[n] e^{-\frac{2\pi j}{N} nk} := \sum_{n=0}^{N-1} x_w[n] W_N^{kn}, \tag{A.2}$$

where $x_w[n] = x[n] \cdot w[n-m]$ denotes the weighted signals to the window function $w[n-m]$, $\tilde{X}_w[k]$ is the result in the frequency domain, $j$ is the imaginary unit and $W_N = e^{-\frac{2\pi j}{N}}$. The essence of the STFT is to apply a Discrete Fourier Transform (DFT) on the resulting windowed signal $x_w[n]$. The DFT formulation in Eq. A.2 can be derived from the Fourier transform for continuous signals. For our method, the STFT plays an essential role in the pre-processing of the acoustic signal.

# B Proofs

## B.1 Proof of Assumption 3.1

*Proof*. Let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be a hierarchical tree, i.e. a directed acyclic graph, where $\mathcal{V}$ is the set of nodes representing possible class labels and $\mathcal{E}$ is the set of edges representing hierarchical relationships among the classes. Each path $\mathcal{P} \subseteq \mathcal{T}$ is a valid path from the root node $v^r = v_1$ to a leaf node $v_c$, with each node $v_i$ being a node in the path.

Since each node $v_i$ depends only on the input $\tilde{x}$ and its preceding node $v_{i-1}$, the probability of the path from the root node $v^r$ to the leaf node $v_c$ can be written as:

$$p(v_1 \to v_2 \to \cdots \to v_c|\tilde{x}) = p(v_1|\tilde{x}) \cdot p(v_2|v_1, \tilde{x}) \cdots p(v_c|v_{c-1}, \tilde{x}). \tag{A.3}$$

Based on conditional independence, we conclude that:

$$p(v_i|v_{i-1}, \tilde{x}) = p(v_i|\tilde{x}). \tag{A.4}$$

By applying Eq. A.3 and Eq. A.4, the path probability formula can be simplified to:

$$\begin{aligned} p(v_1 \to v_2 \to \cdots \to v_c|\tilde{x}) &= p(v_1|\tilde{x}) \cdot p(v_2|\tilde{x}) \cdots p(v_c|\tilde{x}) \\ &= \prod_{i=1}^{|\mathcal{P}|} p(v_i|\tilde{x}). \end{aligned} \tag{A.5}$$

When the hierarchical tree $\mathcal{T}$ consists of only one hierarchy, all leaf nodes are directly connected to the root node, i.e. the path length is 1. In this case, hierarchical inference reduces to a typical classification inference:

$$p(v_1 \to v_c|\tilde{x}) = p(v_1|\tilde{x})p(v_c|v_1, \tilde{x}) = p(v_c|\tilde{x}). \tag{A.6}$$

It follows that traditional classification inference ($cf$. Eq. A.6) is a special case of hierarchical inference ($cf$. Eq. A.5).

## B.2 Derivation of Eq. 5

*Proof*. First, the traditional binary cross-entropy loss function can be written as:

$$\mathcal{L}^{BCE} = \sum_{v \in \mathcal{V}} -\tilde{y}_v \log(s_v) - (1 - \tilde{y}_v) \log(1 - s_v), \tag{A.7}$$

where $s_v$ represents the predicted probability for node $v$ and $y_v \in \{0, 1\}$ denotes the true label for node $v$.

In addition, we impose the following hierarchical constraints:

$$\begin{cases} \hat{s}_v = \min\limits_{u \in \mathcal{V}_A} (s_u) & \tilde{y}_v = 1, \\ 1 - \hat{s}_v = \min\limits_{u \in \mathcal{V}_R} (1 - s_u) & \tilde{y}_v = 0. \end{cases} \tag{A.8}$$

In order to facilitate the incorporation of Eq. A.8 into Eq. A.7, Eq. A.8 is converted as follows:

$$\begin{cases} \hat{s}_v = \min\limits_{u \in \mathcal{V}_A} (s_u) & \tilde{y}_v = 1, \\ \hat{s}_v = 1 - \min\limits_{u \in \mathcal{V}_R} (1 - s_u) = \max\limits_{u \in \mathcal{V}_R} (s_u) & \tilde{y}_v = 0. \end{cases} \tag{A.9}$$

By incorporating the updated prediction scores ($cf$. Eq. A.9) into the binary cross-entropy loss $\mathcal{L}^{BCE}$ ($cf$. Eq. A.7), we obtain the hierarchical tree loss $\mathcal{L}^{HT}$ ($cf$. Eq. **??**):

$$\begin{aligned} \mathcal{L}^{HT} &= \sum_{v \in \mathcal{V}} -\tilde{y}_v \log(s_v) - (1 - \tilde{y}_v) \log(1 - s_v), \\ &= \sum_{v \in \mathcal{V}} -\tilde{y}_v \log(\min_{u \in \mathcal{V}_A} (s_u)) - (1 - \tilde{y}_v) \log(1 - \max_{u \in \mathcal{V}_R} (s_u)). \end{aligned} \tag{A.10}$$

## B.3 Convergence of Eq. 5

*Proof*. To analyse the convergence of the hierarchical tree loss $\mathcal{L}^{HT}$, we need to prove that it satisfies the following five conditions: non-negativity, minimum value of 0, Lipschitz continuity, bounded gradients and deterministic convergence. Next, we will conduct a detailed analysis based on these five aspects.

**Analysis of Non-Negativity.** Each term in the hierarchical tree loss $\mathcal{L}^{HT}$ is non-negative. Specifically:

- When $\tilde{y}_v = 1$, the following holds:

$$-\tilde{y}_v \log(\min_{u \in \mathcal{V}_A} (s_u)) = -\log(\min_{u \in \mathcal{V}_A} (s_u)). \tag{A.11}$$

  Since $s_u \in [0, 1]$, we have $\min\limits_{u \in \mathcal{V}_A} (s_u) \in [0, 1]$. Moreover, $-\log(\cdot)$ is defined over the interval $(0, 1]$, it follows that $-\tilde{y}_v \log(\min\limits_{u \in \mathcal{V}_A} (s_u)) \geq 0$. Therefore, this term is non-negative.

- When $\tilde{y}_v = 0$, the following holds:

$$-(1 - \tilde{y}_v) \log(1 - \max_{u \in \mathcal{V}_R} (s_u)) = -\log(1 - \max_{u \in \mathcal{V}_R} (s_u)). \tag{A.12}$$

  Since $s_u \in [0, 1]$, we have $\max\limits_{u \in \mathcal{V}_R} (s_u) \in [0, 1]$, which implies that $1 - \max\limits_{u \in \mathcal{V}_R} (s_u) \in [0, 1]$. Over this interval, $-\log(1 - \max\limits_{u \in \mathcal{V}_R} (s_u)) \geq 0$. Therefore, this term is also non-negative.

Consequently, the hierarchical tree loss $\mathcal{L}^{HT}$ is non-negative.

**Conditions for Minimum Valve of 0.** For the hierarchical tree loss $\mathcal{L}^{HT}$ to achieve its minimum value of 0, each term must be 0. This requires the following conditions:

- When $\tilde{y}_v = 1$, to satisfy $-\log(\min\limits_{u \in \mathcal{V}_A} (s_u)) = 0$, we need $\min\limits_{u \in \mathcal{V}_A} (s_u) = 1$, which implies that the predicted probability $s_u = 1$ for all ancestor nodes.
- When $\tilde{y}_v = 0$, to satisfy $-\log(1 - \max\limits_{u \in \mathcal{V}_R} (s_u)) = 0$, we need $1 - \max\limits_{u \in \mathcal{V}_R} (s_u) = 1$, i.e. $\max\limits_{u \in \mathcal{V}_R} (s_u) = 0$. This implies that the predicted probability $s_u = 0$ for all child nodes.

Based on the above, when these conditions are met, the hierarchical tree loss $\mathcal{L}^{HT} = 0$.

**Analysis of Lipschitz continuity.** First, we decompose the function $\mathcal{L}^{HT}$ into two parts:

$$\mathcal{L}^{HT} = \underbrace{-\tilde{y}_v \log(\min_{u \in \mathcal{V}_A} s_u)}_{:= f_1(\theta)} + \underbrace{-(1 - \tilde{y}_v) \log(1 - \max_{u \in \mathcal{V}_R} s_u)}_{:= f_2(\theta)}. \tag{A.13}$$

Then, it suffices to prove that the gradients of $f_1$ and $f_2$ are Lipschitz continuous, respectively.

- Analysis of the gradient of $f_1(\theta) = -\tilde{y}_v \log(\min\limits_{u \in \mathcal{V}_A} s_u)$. Assuming $s_u = \sigma(z_u)$ ($\sigma$ is the sigmoid function) and $z_u = w_u^T h + b_u$ ($h$ is the hidden layer output). According to the chain rule, we have:

$$\nabla_{w_u} f_1 = -\frac{1}{\min s_u} \cdot \mathbb{1}(s_u = \min s_u) \cdot \sigma'(z_u) \cdot h, \tag{A.14}$$

  where $\mathbb{1}(\cdot)$ is an indicator function. Since $\sigma'(z_u) = s_u(1 - s_u) \leq \frac{1}{4}$ and after numerical stabilization $\min s_u \geq \varepsilon$, we obtain:

$$\left\| \nabla_{w_u} f_1 \right\| \leq \frac{1}{\varepsilon} \cdot \frac{1}{4} \cdot \|h\|. \tag{A.15}$$

Since the hidden layer output $h$ is bounded, then $\nabla f_1$ is bounded. According to the boundedness of the gradient and the mean value theorem, for any $\theta_1$ and $\theta_2$, we have:

$$\|\nabla f_1(\theta_1) - \nabla f_1(\theta_2)\| \leq \alpha_1 \|\theta_1 - \theta_2\|, \tag{A.16}$$

where $\alpha$ depends on $\varepsilon$ and the Lipschitz constant of the network parameters.

- Analysis of the gradient of $f_2(\theta) = -(1-\tilde{y}_v)\log(1-\max_{u\in V_R} s_u)$. Similarly, following the same reasoning as above, the Lipschitz constant $\alpha_2$ of $\nabla f_2$ is determined by the max operator and the boundedness of the gradient of $1 - s_u$.

By setting $\alpha = \alpha_1 + \alpha_2$, we conclude that the gradient of $\mathcal{L}^{HT}$ is Lipschitz continuous.

**Analysis of Gradient Bound.** The gradient boundary of $\mathcal{L}^{HT}$ is composed of the bounding of $\nabla f_1$ and the bounding of $\nabla f_2$.

- Analysis of bounding $\nabla f_1$. Assuming $\|h\| \leq c$, based on Eq. A.15, we have:

$$\|\nabla f_1\| \leq \frac{c}{4\varepsilon}. \tag{A.17}$$

- Analysis of bounding $\nabla f_2$. Similarly, we have:

$$\|\nabla f_2\| \leq \frac{c}{4\varepsilon}. \tag{A.18}$$

Since $\mathcal{L}^{HT} = f_1 + f_2$ and $C = \frac{c}{2\varepsilon}$, we have $\left\|\nabla \mathcal{L}^{HT}\right\| \leq C$. Therefore, the gradient of $\mathcal{L}^{HT}$ is bounded.

**Analysis of Deterministic Convergence.** Since $\nabla \mathcal{L}^{HT}$ is Lipschitz continuous, for any $\theta$ and $\theta'$, we have:

$$\mathcal{L}^{HT}(\theta') \leq \mathcal{L}^{HT}(\theta) + \nabla \mathcal{L}^{HT}(\theta)^T(\theta' - \theta) + \frac{\alpha}{2}\|\theta' - \theta\|^2. \tag{A.19}$$

Based on gradient descent update $\theta' = \theta - \eta \nabla L^{HT}(\theta)$, we have:

$$\mathcal{L}^{HT}(\theta_{k+1}) \leq \mathcal{L}^{HT}(\theta_k) - \eta(1 - \frac{\eta\alpha}{2})\left\|\nabla \mathcal{L}^{HT}(\theta_k)\right\|^2. \tag{A.20}$$

When $\eta < \frac{2}{\alpha}$, then $\mathcal{L}^{HT}(\theta_{k+1}) < \mathcal{L}^{HT}(\theta_k)$, i.e. strictly decreasing. Since $\mathcal{L}^{HT}$ is bounded, then the monotonically decreasing sequence $\{L^{HT}(\theta_k)\}$ converges to a certain value $\alpha^*$. Next, the sum of Eq. A.20, we have:

$$\sum_{k=0}^{\infty} \eta(1 - \frac{\eta\alpha}{2})\left\|\nabla \mathcal{L}^{HT}(\theta_k)\right\|^2 \leq \mathcal{L}^{HT}(\theta_0) - \alpha^* < \infty. \tag{A.21}$$

Therefore, we have $\lim_{k\to\infty} \left\|\nabla L^{HT}(\theta_k)\right\| = 0$, i.e. $\mathcal{L}^{HT}$ satisfies deterministic convergence.

In summary, the hierarchical tree loss $\mathcal{L}^{HT}$ is convergent. Similarly, the focal hierarchical tree loss $\mathcal{L}^{FHT}$ is also convergent.

## B.4 Differentiability Analysis of Eq. 5

*Proof.* The min and max operations in Eq. 5 can introduce non-differentiable points due to sudden changes in output as internal node scores change. Specifically, the results of the min and max operations may shift abruptly as the values of the internal nodes vary, leading to non-differentiable points. However, in deep learning optimization, we can handle such non-differentiable cases using subgradients or alternative methods.

- **Smooth Approximation:** The min and max operators can be approximated with differentiable smooth functions, such as the Softplus function or the LogSumExp function, as follows:

$$\begin{aligned} \min(x_1, x_2, \cdots, x_n) &\approx -\frac{1}{\alpha}\log(\sum_{i=1}^{n} e^{-\alpha x_i}), \\ \max(x_1, x_2, \cdots, x_n) &\approx \frac{1}{\alpha}\log(\sum_{i=1}^{n} e^{\alpha x_i}), \end{aligned} \tag{A.22}$$

where $\alpha$ is a smoothing parameter and as $\alpha \to \infty$, the approximation becomes increasingly accurate.

- **Gradient Computation:** In practical optimization, even if non-differentiable points exist, automatic differentiation frameworks (e.g. PyTorch and TensorFlow) can generally handle the gradients of piecewise smooth function, enabling parameter updates to proceed without obstruction.

Although the hierarchical tree loss $\mathcal{L}^{HT}$ contains min and max operations that lead to non-differentiable points, it can still be optimized using smooth approximations or subgradient methods.

Specifically, considering the gradient derivation of the min and max parts in the hierarchical tree loss $\mathcal{L}^{HT}$ ($cf.$ Eq. 5):

- When $\tilde{y}_v = 1$, the loss term is:

$$-\tilde{y}_v \log(\min_{u\in\mathcal{V}_A}(s_u)) = -\log(\min_{u\in\mathcal{V}_A}(s_u)). \tag{A.23}$$

Using the LogSumExp approximation as follows:

$$-\log(\min_{u\in\mathcal{V}_A}(s_u)) \approx \frac{1}{\alpha}\log(\sum_{u\in\mathcal{V}_A} e^{-\alpha s_u}). \tag{A.24}$$

The gradient of the loss function is:

$$\frac{\partial \mathcal{L}^{HT}}{\partial s_u} = -\frac{e^{-\alpha s_u}}{\sum_{u\in\mathcal{V}_A} e^{-\alpha s_u}}. \tag{A.25}$$

This is a smooth and differentiable gradient.

- When $\tilde{y}_v = 0$, the loss term is:

$$-(1-\tilde{y}_v)\log(1-\max_{u\in\mathcal{V}_R}(s_u)) = -\log(1-\max_{u\in\mathcal{V}_R}(s_u)). \tag{A.26}$$

Applying the LogSumExp approximation as follows:

$$-\log(1-\max_{u\in\mathcal{V}_R}(s_u)) \approx \frac{1}{\alpha}\log(\sum_{u\in\mathcal{V}_R} e^{\alpha s_u}). \tag{A.27}$$

The gradient of the loss function is:

$$\frac{\partial \mathcal{L}^{HT}}{\partial s_u} = -\frac{e^{\alpha s_u}}{\sum_{u\in\mathcal{V}_R} e^{\alpha s_u}}. \tag{A.28}$$

This is a smooth and differentiable gradient.

In summary, the min and max operations in Eq. 5 are generally non-differentiable. However, we can replace them with differentiable expressions, ensuring the loss function is differentiable over the entire domain. This smooth approximation stabilizes gradient computation, allowing effective gradient calculation in deep learning through automatic differentiation tools.

In many cases, loss functions contain min or max operations, which are often introduced to incorporate some form of non-linearity or heuristic constraints. Although these operations can lead to points of non-differentiability in the loss function, they remain important in deep learning and optimization, as follows:

- **Introducing Discontinuities:** The max or min operations can cause the slope of the function to change abruptly at certain points, leading to non-differentiable points. In other words, the loss function can change its shape under specific conditions when the predicted value equals a threshold.
- **Enhancing Model Capability:** By maximizing or minimizing certain quantities, loss functions can better capture complex patterns in the data, which can guide the model to learn more effectively.
- **Improving Model Robustness:** The max or min operations make the model more robust to noise and uncertainty in the input data.
- **Incorporating Prior Knowledge:** The max or min operations can enforce prior knowledge or constraints.

In summary, although max or min operations may introduce non-differentiable points in the loss function, their advantages and effectiveness in the optimization process often outweigh this drawback. In practice, many optimization algorithm (e,g, Adam and RMSProp, etc.) can handle these discontinuities effectively, resulting in good training results.

## B.5 Proof of Triangle Inequality for $\psi(\cdot,\cdot)$

*Proof.* For any three nodes $u$, $v$, $w$ in a hierarchical tree $\mathcal{T}$, the distance between any two nodes $u$ and $v$ is defined as:

$$\psi(u,v) = \psi(u, lca(u,v)) + \psi(v, lca(u,v)), \tag{A.29}$$

where $\psi(u,v)$ represents the path length between nodes $u$, $v$ and $lca(u,v)$ denotes the lowest common ancestor of $u$ and $v$. Now, consider any three nodes $u, v, w \in \mathcal{T}$ and define their pairwise lowest common ancestors:

$$\begin{cases} L_1 = lca(u,v) \\ L_2 = lca(v,w) \\ L_3 = lca(u,w) \end{cases}. \tag{A.30}$$

We can express the path from node $u$ to node $w$ as:

$$\psi(u,w) = \psi(u,L_3) + \psi(L_3,w), \tag{A.31}$$

where $L_3 = lca(u,w)$ is the starting point of the shared path between $u$ and $w$. Based on the properties of lowest common ancestors, the path from $u$ to $L_3$ can be decomposed as:

$$\psi(u,L_1) + \psi(L_1,L_2) + \psi(L_2,w). \tag{A.32}$$

Thus, we have:

$$\begin{aligned} \psi(u,w) &= \psi(u,L_3) + \psi(w,L_3) \\ &\le (\psi(u,L_1) + \psi(L_1,L_2) + \psi(L_2,w)) + \psi(w,L_3). \end{aligned} \tag{A.33}$$

Eq. A.33 can be further expressed as:

$$\begin{aligned} \psi(u,w) &\le \psi(u,L_1) + \psi(v,L_1) + \psi(v,L_2) + \psi(w,L_2) \\ &= \psi(u,v) + \psi(v,w). \end{aligned} \tag{A.34}$$

Therefore, the tree distance $\psi(\cdot,\cdot)$ satisfies the triangle inequality for any three nodes $u, v, w \in \mathcal{T}$, which proves the validity of this distance measure.

## B.6 Proof of $m_\sigma$ Boundary

*Proof.* To determine the boundary of $m_\sigma = \frac{\psi(\hat{v}_c,\hat{v}_c^-) - \psi(\hat{v}_c,\hat{v}_c^+)}{2H}$, we need to calculate the maximum distance $\max \psi(\hat{v}_c,\hat{v}_c^+)$ and minimum distance $\min \psi(\hat{v}_c,\hat{v}_c^+)$ between the anchor sample $i$ and the positive sample $i^+$, as well as the maximum distance $\max \psi(\hat{v}_c,\hat{v}_c^-)$ and minimum distance $\min \psi(\hat{v}_c,\hat{v}_c^-)$ between the anchor sample $i$ and the negative sample $i^-$, respectively.

**Distance between anchor and positive samples.** Given the anchor sample $i$ and the positive sample $i^+$ are located at their respective leaf nodes $\hat{v}_c$ and $\hat{v}_c^+$ and that they satisfy the sample selection strategy $g_c(i) = g_s(i^+)$ ($cf.$ Def. 3.7), the minimum distance $\min \psi(\hat{v}_c,\hat{v}_c^+)$ between the anchor and positive samples occurs only when both samples are located at the same leaf node. In other words, if the anchor sample $i$ and positive sample $i^+$ both reside at node $\hat{v}_c$, then we have:

$$\min \psi(\hat{v}_c,\hat{v}_c^+) = \psi(\hat{v}_c,\hat{v}_c) = \psi(\hat{v}_c^+,\hat{v}_c^+) = 0. \tag{A.35}$$

For the maximum distance $\max \psi(\hat{v}_c,\hat{v}_c^+)$, we consider the scenario where the anchor sample $i$ and the positive sample $i^+$ share their lowest common ancestor node $u$. The path between $\hat{v}_c$ and $\hat{v}_c^+$ in the tree can be represented as $\hat{v}_c \to u \to \hat{v}_c^+$. Consequently, the maximum distance is given by:

$$\max \psi(\hat{v}_c,\hat{v}_c^+) = \psi(\hat{v}_c,u) + \psi(u,\hat{v}_c^+) = 1 + 1 = 2. \tag{A.36}$$

Therefore, the maximum value of $\psi(\hat{v}_c,\hat{v}_c^+)$ is 2 and the minimum value of $\psi(\hat{v}_c,\hat{v}_c^+)$ is 0, i.e. $\max \psi(\hat{v}_c,\hat{v}_c^+) = 2$ and $\min \psi(\hat{v}_c,\hat{v}_c^+) = 0$. An example is shown in Figure A2.

**Distance between anchor and negative samples.** For the minimum distance $\min \psi(\hat{v}_c,\hat{v}_c^-)$, given $u$ is the parent node of node $\hat{v}_c$ corresponding to the anchor sample $i$, $\dot{u}$ is the sibling node of $u$ and $\ddot{u}$ is the ancestor nodes of both $u$ and $\dot{u}$. The anchor sample $i$ and negative sample $i^-$ can only come from their leaf nodes $\hat{v}_c$ and $\hat{v}_c^-$. Moreover, they satisfy the sampling strategy $g_c(i) \neq g_s(i^-)$ ($cf.$ Def. 3.7). The path from $\hat{v}_c$ to $\hat{v}_c^-$ can be described as $\hat{v}_c \to u \to \dot{u} \to \ddot{u}$, i.e. $\hat{v}_c \to u \to \dot{u} \to \hat{v}_c^-$. Therefore, we have:

$$\min \psi(\hat{v}_c,\hat{v}_c^-) = \psi(\hat{v}_c,u) + \psi(u,\dot{u}) + \psi(\dot{u},\hat{v}_c^-) = 1+1+1 = 3. \tag{A.37}$$

In other words, a complete binary tree is a necessary condition for achieving the minimum distance between the anchor sample node and the negative sample node.

For the maximum distance $\max \psi(\hat{v}_c,\hat{v}_c^-)$, since the anchor sample $i$ and the negative sample $i^-$ can only come from the corresponding leaf nodes $\hat{v}_c$ and $\hat{v}_c^-$. In addition, they must also satisfy $g_c(i) \neq g_s(i^-)$ ($cf.$ Def. 3.7). The distance $\psi(\hat{v}_c,\hat{v}_c^-)$ is maximized only when the anchor sample node $\hat{v}_c$ and the negative sample node $\hat{v}_c^-$ are located in different subtrees of depth $H$, as follows:

$$\max \psi(\hat{v}_c,\hat{v}_c^-) = H + H = 2H. \tag{A.38}$$

Therefore, the maximum value of $\psi(\hat{v}_c,\hat{v}_c^-)$ is $2H$ and the minimum value of $\psi(\hat{v}_c,\hat{v}_c^-)$ is 3, i.e. $\max \psi(\hat{v}_c,\hat{v}_c^-) = 2H$ and $\min \psi(\hat{v}_c,\hat{v}_c^-) = 3$. An example is shown in Figure A2.

Based on the above, the minimum value of $m_\sigma$ is as follows:

$$\begin{aligned}\min m_\sigma &= \min \frac{\psi(\hat{v}_c, \hat{v}_c^-) - \psi(\hat{v}_c, \hat{v}_c^+)}{2H} \\ &= \frac{\min \psi(\hat{v}_c, \hat{v}_c^-) - \max \psi(\hat{v}_c, \hat{v}_c^+)}{2H} \\ &= \frac{1}{2H}.\end{aligned} \tag{A.39}$$

Similarly, the maximum value of $m_\sigma$ can be expressed as follows:

$$\begin{aligned}\max m_\sigma &= \max \frac{\psi(\hat{v}_c, \hat{v}_c^-) - \psi(\hat{v}_c, \hat{v}_c^+)}{2H} \\ &= \frac{\max \psi(\hat{v}_c, \hat{v}_c^-) - \min \psi(\hat{v}_c, \hat{v}_c^+)}{2H} \\ &= \frac{2H}{2H} = 1.\end{aligned} \tag{A.40}$$

When $H \to \infty$, we have $\lim(\min m_\sigma) = \lim \frac{1}{2H} \to 0$ and $\lim(\max m_\sigma) = 1$. Therefore, we have $m_\sigma \in (0, 1]$.

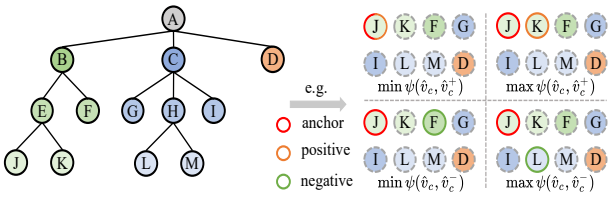



**Figure A2: Schematic diagram of the maximum and minimum boundaries for anchor sample nodes, positive sample nodes and negative sample nodes. The left part shows a given hierarchical tree and the right part provides examples of maximum and minimum distances.**

# C Method

## C.1 Framework of Model

The overall architecture of our model is depicted in Figure A3. Given a signal dataset $\mathcal{X} = \{X_i, i = 1, 2, \ldots, N\} \subseteq \mathbb{R}^{M \times N}$ and the corresponding label $\mathcal{Y} \subseteq \mathbb{R}^{C \times N}$ with $N$ streams, $M$ measurements for each stream and $C$ fault classes. For feature representation learning, the signal dataset $\mathcal{X}$ is input into the acoustic signals preprocessing module and $\tilde{\mathcal{X}} \subseteq \mathbb{R}^{T \times F \times 3}$ is the output after the sliding window (SW) and STFT operations, as follows:

$$\begin{aligned}\tilde{\mathcal{X}} &= 10 \times \log_{10}\left(\frac{|\mathrm{STFT}(\mathrm{SW}(\mathcal{X}))|}{\max(|\mathrm{STFT}(\mathrm{SW}(\mathcal{X}))|)}\right) \\ &:= 10 \times \log_{10}\left(\frac{|\mathcal{X}_{sw}[n,m]|}{\max(|\mathcal{X}_{sw}[n,m]|)}\right),\end{aligned} \tag{A.41}$$

where $\mathcal{X}_{sw}[n, m]$ indicates the $n$-th row and $m$-th column elements of the STFT result matrix and $|\mathcal{X}_{sw}[n, m]|$ denotes the elements of the amplitude spectrum matrix. Then, $\tilde{\mathcal{X}}$ is fed into the feature learning module (FL) to produce learned features $\boldsymbol{F} \in \mathbb{R}^D$ with $D$ denoting the dimension of the T-F domain spectrogram. Finally, the hierarchical classification score $\boldsymbol{s} \in \mathcal{V} \in [0, 1]^{|\mathcal{V}|}$ cen be computed by $\boldsymbol{s} = \mathrm{FL}(\tilde{\mathcal{X}})$. The whole process is trained and optimized through a training objective, i.e. $\mathcal{L} = \frac{h_i}{\sum_i h_i} \times \mathcal{L}^{FHT} + \alpha \mathcal{L}^{GTT}$, consisting of the focal hierarchical tree loss $\mathcal{L}^{FHT}$ ($cf.$ Eq. **??**) and the group tree triplet loss $\mathcal{L}^{GTT}$ ($cf.$ Eq. **??**).

# D Experiments

## D.1 Evaluation Metrics

As mentioned in the evaluation metrics, we apply dynamic thresholds to assess the performance of fault intensity diagnosis for the proposed HKG model. For any given threshold, we can determine the counts of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). In addition, the following metrics are calculated:

$$\begin{aligned}\text{Accuracy} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \\ \text{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \\ \text{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \\ \text{F1-score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.\end{aligned} \tag{A.42}$$

By evaluating across all possible thresholds, we can generate a precision-recall curve, with precision plotted on the $y$-axis and recall on the $x$-axis. The Average Precision (AP) is calculated as $AP = \sum_n \left(\frac{R_n - R_{n-1}}{P_n}\right)$, where $P_n$ and $R_n$ represent the precision and recall at the $n$-th threshold, respectively.

## D.2 Datasets

*D.2.1 Cavitation Datasets.* The cavitation datasets are provided by SAMSON AG in Frankfurt. The schematic of the experimental setup is illustrated in Figure A4. The acoustic signals are recorded under five distinct flow conditions, generated by adjusting the differential pressure at various constant upstream pressures of the control valve. These conditions include choked flow cavitation, constant cavitation, incipient cavitation, turbulent flow, and no flow (see Table A1 and Table A2). The detailed dataset statistics and label distributions of three real-world cavitation datasets without data augmentation are provided in Table A3. In addition, the hierarchical tree of cavitation datasets is shown in Figure A5.

**Table A1: The content details of the three real-world cavitation datasets for each flow state.**

| Dataset | Cavitation | | | Non-cavitation | |
|---|---|---|---|---|---|
| | choked flow | constant | incipient | turbulent | no flow |
| Cavitation-Short | 72 | 93 | 40 | 118 | 33 |
| Cavitation-Long | 148 | 396 | 64 | 183 | 15 |
| Cavitation-Noise | 40 | 40 | 40 | 40 | 0 |

**Table A2: Details of three real-world cavitation datasets for valve operation with various upstream pressures.**

| Dataset | Operation parameters | | |
|---|---|---|---|
| | Valve stroke (mm) | Upstream pressure (bar) | Temperature (°C) |
| Cavitation-Short | [15,13.5,11.25,7.5,3.75,1.5,0.75] | [10,9,6,4] | 25-50 |
| Cavitation-Long | [60,55,45,30,25,15,6] | [10,6,4] | 23-52 |
| Cavitation-Noise | 15 | 10 | 32-39 |

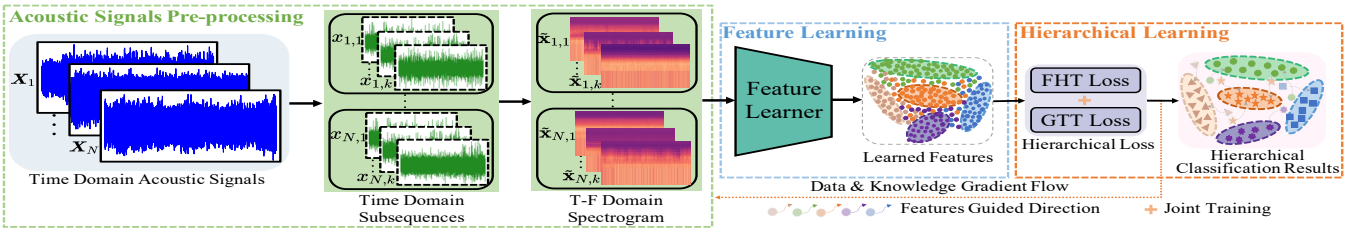



**Figure A3: Overall framework of the DHK. The T-F domain spectrograms are fed into feature representation learning module to extract deep learned features. Meanwhile, the features are jointly trained and optimized by FHT and GTT loss functions for hierarchical fault intensity recognition.**

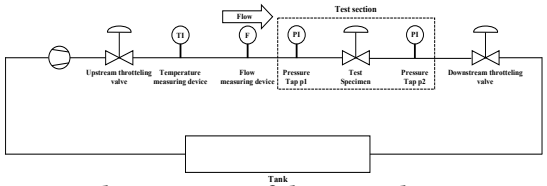



**Figure A4: Schematic view of the test rack at SAMSON AG.**

**Table A3: Details of the training and test sets. $(\cdot)$ denotes the number after the sliding window (window size is 466944).**

| Dataset | Training set | | | | Testing set | | | |
|---|---|---|---|---|---|---|---|---|
| | Cavitation | | | Non | Cavitation | | | Non |
| | Choked flow | Constant | Incipient | | Choked flow | Constant | Incipient | |
| Cavitation-Short | 58(×10) | 75(×10) | 32(×10) | 121(×10) | 14(×10) | 18(×10) | 8(×10) | 30(×10) |
| Cavitation-Long | 118(×83) | 317(×83) | 52(×83) | 158(×83) | 30(×83) | 79(×83) | 12(×83) | 40(×83) |
| Cavitation-Noise | 32(×83) | 32(×83) | 32(×83) | 32(×83) | 8(×83) | 8(×83) | 8(×83) | 8(×83) |

*D.2.2 PUB Dataset.* This dataset is used to evaluate the scalability of our method. The levels of bearing damage are detailed in Table A4. The file codes and corresponding fault types used in our experiment are provided in Table A5. The PUB is organized into three hierarchies: bearing diagnosis (Hierarchy I), bearing damage type diagnosis (Hierarchy II) and IR/OR intensity diagnosis (Hierarchy III-IR/III-OR), as depicted in Figure A6.

**Table A4: The bearing fault damage levels in the PUB.**

| Damage level | Percentage values | Bearing limitations |
|---|---|---|
| 1 | 0-2% | ≤2 mm |
| 2 | 2-5% | >2 mm |
| 3 | 5-15% | >4.5 mm |

## D.3 Results

*D.3.1 Confusion Matrix.* We show the confusion matrix of the best performance backbone and our method on four real-world datasets, as shown in Figure A7.

**Cavitation Datasets:** It can be seen that our method can significantly improve the performance of each cavitation state, especially the incipient cavitation state. Specifically, DHK+Uniformer-B outperforms UniFormer-B by **7.5**%, **0.3**% and **2.25**% for the incipient cavitation state on three different cavitation datasets, respectively.

**Table A5: The bearing fault types and file codes in the PUB.**

| Fault type | Healthy | OR damage | | IR damage | | |
|---|---|---|---|---|---|---|
| | | OR-1 | OR-2 | IR-1 | IR-2 | IR-3 |
| File code | K001 | KA01 | KA03 | KI01 | KI07 | KI16 |
| | K002 | KA05 | KA06 | KI03 | KI08 | - |
| | K003 | KA04 | KA08 | KI04 | KI18 | - |
| | K004 | KA07 | KA09 | KI05 | - | - |
| | K005 | KA15 | KA16 | KI14 | - | - |
| | K006 | KA22 | - | KI17 | - | - |
| | - | KA30 | - | KI21 | - | - |

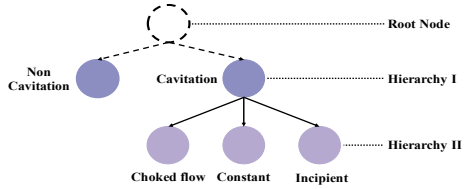



**Figure A5: Hierarchical cavitation tree from cavitation datasets.**

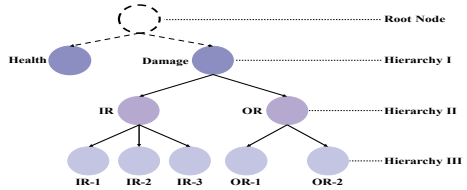
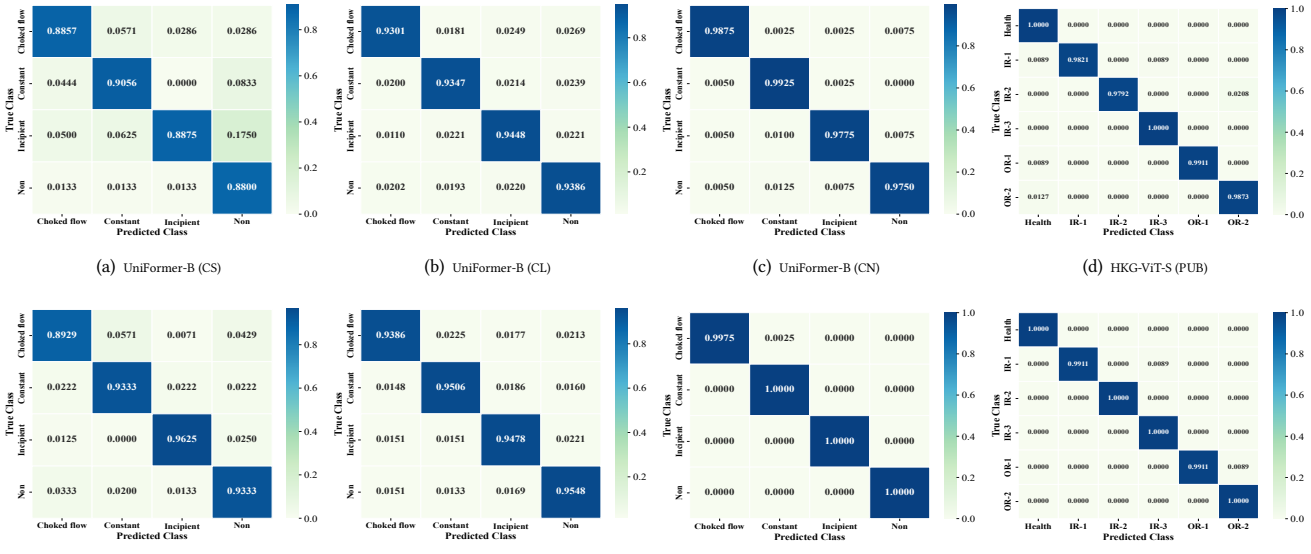



**Figure A6: Hierarchical bearing fault tree from the PUB dataset.**

**PUB Dataset:** DHK+PerViT-B achieves **100**% accuracy in health, IR-2, IR-3 and OR-2. Moreover, DHK+PerViT-B reaches **99.11**% accuracy on both IR-1 and OR-1.

*D.3.2 Computational Complexity for DHK.* Compared to $\mathcal{L}^{CCE}$ and $\mathcal{L}^{Focal}$, the proposed DHK introduces hierarchical constraints and a group triplet loss, resulting in minimal additional computational complexity. Notably, unlike methods dependent on hierarchical feature extraction modules or specially designed architectures, the proposed DHK significantly reduces implementation and computational costs without requiring additional hierarchical feature extraction modules or knowledge-embedded structures. Therefore,




(a) UniFormer-B (CS) (b) UniFormer-B (CL) (c) UniFormer-B (CN) (d) HKG-ViT-S (PUB)

(e) DHK+UniFormer-B (CS) (f) DHK+UniFormer-B (CL) (g) DHK+UniFormer-B (CN) (h) DHK+PerViT (PUB)

**Figure A7: The confusion matrix of DHK with different backbone networks on various datasets. (a)-(c) and (e)-(g) denote the confusion matrix on Cavitation-Short (CS), Cavitation-Long (CL) and Cavitation-Noise (CN), respectively. (d) and (h) represent the confusion matrix on the PUB dataset.**

we analyse the computational complexity and efficiency of DHK from theoretical and experimental angles:

- $\mathcal{L}^{CCE}/\mathcal{L}^{Focal}$: The computational complexity of $\mathcal{L}^{CCE}$ is $O(NC)$ ($N$: number of samples, $C$: number of classes). While $\mathcal{L}^{Focal}$ only adds a weighted factor, which cannot increase complexity, i.e. the computational complexity is also $O(NC)$.
- DHK Loss: The DHK loss composed of $\mathcal{L}^{HT}/\mathcal{L}^{FHT}$ and $\mathcal{L}^{GTT}$. $\mathcal{L}^{HT}$ computes hierarchical constraints with computational complexity $O(NH)$ ($H$: depth of tree). $\mathcal{L}^{FHT}$ adds a weighted factor and cannot increase complexity, i.e. the computational complexity is also $O(NH)$. $\mathcal{L}^{GTT}$ computes inter-class distances with computational complexity $O(T)$ ($T$: number of valid triplets). Based on the above, the overall computational complexity of DHK loss is $O(NH + T)$.
- Experimental Evaluation: We conducted an experimental comparison of the training time for ResNet18 with $\mathcal{L}^{CCE}$ and ResNet18 with DHK under the same conditions on Cavitation-Short, as shown in Table 10. It can be seen that DHK loss only introduces a minimal training time cost compared to CCE loss.

In summary, although DHK loss has a slightly higher computational complexity compared CCE and Focal loss, we believe that the performance improvement brought by hierarchical optimization strategy far outweigh the minor additional computational cost.

*D.3.3 Inference Time for Eq. 3.* During inference, Eq. 3 calculates each event score from the root to the leaf path. In fact, the essence of Eq. 3 is a greedy algorithm. Therefore, we analyse Eq. 3 from both theoretical and experimental aspects, as follows:

- Computational Complexity Analysis: The computational cost of Eq. 3 mainly depends on the depth $H$ of hierarchical label tree $\mathcal{T}$ and the branching factor at each node. Since the inference process only requires computing the accumulated score along a single optimal path, the time complexity of this method is approximately $O(H)$.
- Comparison with Traditional Classification Methods: In typical flat classification, the model typically computes probabilities for all possible classes and uses softmax normalization, resulting in time complexity of $O(C)$ ($C$: number of classes). In contrast, Eq. 3 reduces the search space by path constraints and the computational cost is reduced when the $H \ll C$. In general, the depth of $\mathcal{T}$ for fault signal or natural image datasets (e.g. ImageNet, CIFAR, ANIMAL-10N and Pathology) is less than the number of subnodes.
- Experimental Evaluation: We calculate the sample inference time on Cavitation-Short, as shown in Table 11. The server is equipped with 24GB RAM, a 12-core CPU and an NVIDIA RTX 2070. The test dataset is 600 and batch size is 8. From Table 11, It can clearly be seen that there is almost no difference in the inference time between DHK and CCE.

# E Discussion

In this section, we reflect on the key assumptions underlying our method (DHK), discuss its limitations, extensibility and consider the broader impact of our work. In addition, we also outline potential directions for future improvements to enhance the effectiveness and applicability of our approach.

## E.1 Assumption

In this paper, we assume that the labels of the dataset (three real-world cavitation dataset provided by SAMSON AG and one publicly available bearing dataset) are **clean and accurate**, i.e. all samples are are annotated with **true and noise-free labels**. Based on this

assumption, the hierarchical label tree (Figure A5 and Figure A6) constructed based on dataset labels can truly reflect the **semantic relationship** and **hierarchical structure information** among target classes. Therefore, this hierarchical structure information is regarded as reliable prior knowledge throughout the model design and training process.

## E.2 Limitations

**Manually Build Hierarchical Label Tree.** In this study, the hierarchical label trees (Figure A5 and Figure A6) are manually constructed based on semantic relationships among target classes and domain-specific prior knowledge. In some cases, manually building a hierarchical label tree can more accurately capture the hierarchical information between target classes, particularly when dealing with a moderate number classes and clearly defined hierarchical structures. In addition, even for the same dataset, the organization of hierarchical label tree may be different when facing different task objectives. At this point, the manual construction method has obvious advantages in terms of flexibility and task adaptability. However, the manual approach also suffers from a certain degree of subjectivity, high construction costs and limited scalability when applied to large-scale class systems or cross-domain tasks.

**Label Noise.** In practical applications, label noise is inevitable and it mainly originates from annotation errors, data ambiguity or sensor inaccuracies. These noises may cause the model to incorrectly learn class relationships, which affects its generalization ability and performance. Although this study assumes that the labels in the dataset are accurate, label noise remains an important challenge for future research. It is worth noting that our method (DHK) relies on the relative hierarchical relationships among classes, which to some extent enhances the robustness for label noise (see Table 12). However, label noise may still interfere with the learning of hierarchical relationships and impact model performance. In most cases, the proportion of label noise among all labels is extremely small and its impact on the overall model is negligible. Nevertheless, from a scientific rigor standpoint, this issue still needs to be addressed.

## E.3 Extensibility

In this study, although the proposed method (DHK) focuses on applications within complex industrial systems (cavitation intensity diagnosis industrial system and bearing strength diagnosis industrial system), the proposed hierarchical tree loss with two adaptive weighting schemes ($cf$. Eq. 5 / Eq. 7 w/ Eq. 6) and group tree triplet loss with a hierarchical dynamic margin($cf$. Eq. 8 w/ Eq. 9) are novel loss functions designed for general tree structures. They are specifically developed for the general hierarchical classification task and have significant theoretical innovation. Based on this, our proposed method in this study can also be effectively extended to other hierarchical classification scenarios, as follows:

- **Natural Image Classification:** In natural images, classes often exhibit semantic hierarchical relationships (e.g. animal → mammal → dog → golden retriever). The model makes rational use of hierarchical information to improve classification performance and robustness. The hierarchical natural image classification datasets include ImageNet, CIFAR and ANIMAL-10N, etc. To exhibit the extensibility of DHK, we have extended experiments on the natural image dataset ANIMAL-10N, see Table A6.
- **Medical Disease Grading:** Medical diagnostic tasks often involve grading systems (e.g. diabetic retinopathy is organized into five grades), where the grades exhibit correlations and progression order. Therefore, hierarchical modeling can more accurately reflect the disease progression path and enhance the model's ability to support clinical decision.
- **Protein Function Prediction:** Protein functions are typically organized into a multi-hierarchy system based on the Gene Ontology structure, where there are significant parent-child relationships among various functions. Therefore, introducing hierarchical information into the model can help enhance its ability to integrate biological priors and improve the biological plausibility of predictions.
- **Scene Recognition:** In scene recognition tasks, classes such as "traffic scene" can be organized into "highway", "city road" and "rural road". By leveraging the hierarchical relationships among scene classes can guide the model to understand scene semantics from coarse to fine levels, improving recognition stability in complex environments.

For these tasks, the hierarchical organization and relationship of the target classes have a significant impact on the model performance. Therefore, our proposed method (DHK) provides a hierarchical structure modeling and optimization strategy that can effectively enhance the model's performance.

**Table A6: Results for DHK on ANIMAL-10N.**

| Methods | Year | Accuracy |
|---|---|---|
| SEFLIE | 2019 | 81.8 |
| DivideMix | 2020 | 84.5 |
| PLC | 2021 | 83.4 |
| DAL | 2023 | 82.66 |
| BR | 2023 | 85.8 |
| OT-Filter | 2023 | 85.5 |
| LongReMix | 2023 | 86.88 |
| C2MT | 2024 | 85.8 |
| CFNL | 2024 | 84.5 |
| VRI | 2024 | 85.8 |
| DHK | 2025 | 85.92 |

## E.4 Broader Impact

Our proposed DHK method whose goal is to advance the field of "AI + Industry". The DHK is designed to enhance the performance of fault intensity diagnosis and cavitation intensity recognition, which are crucial for ensuring the reliability and efficiency of industrial systems. In addition, the DHK enhances model performance and interpretability in industrial fault diagnosis tasks, potentially alleviating the workload of monitors in complex industrial systems. Its strong adaptability to hierarchical classification tasks also makes DHK suitable for deployment in high-risk domains. We advocate for a cautious and responsible attitude in real-world applications to ensure the method's reliability and compliance in terms of fairness, transparency, and safety.

## E.5 Future Improvements

Although the DHK performs excellently in industrial fault diagnosis, there is still several aspects that warrant further improvement and exploration in the future, as outlined below:

- **Label Noise Handling:** Although most datasets contain a relatively low proportion of noisy labels, label noise can still affect the training performance of the model. In the future, more robust algorithms can be investigated into our proposed method DHK to better deal with noisy labels (e.g. co-teaching, label smoothing and self-supervised learning, etc.), aiming to achieve the model stability and accuracy even in the presence of inconsistent or incorrect labels.
- **Dynamic Hierarchical Structure Modeling:** The hierarchical label tree in this study assumes that the class relationships are fixed and explicitly defined. However, fault patterns and system states may evolve over time in real-world industrial applications. Future work can explore more flexible hierarchical tree modeling approaches (e.g. graph neural network, structure learning mechanisms and time-aware graph modeling techniques, etc.) to dynamically adjust the hierarchical structure, enabling the model to adapt to evolving system states and fault patterns.