Is the Information Bottleneck Robust Enough? Towards Label-Noise Resistant Information Bottleneck Learning

Yi Huang[1], Qingyun Sun[1*], Yisen Gao[2], Haonan Yuan[1], Xingcheng Fu[3], Jianxin Li[1]

[1]SKLCCSE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Department of Computer Science and Engineering, HKUST, Hong Kong, China
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
{yihuang, sunqy, yuanhn, lijx}@buaa.edu.cn, ygaodi@cse.ust.hk, fuxc@gxnu.edu.cn

Abstract

The Information Bottleneck (IB) principle facilitates effective representation learning by preserving label-relevant information while compressing irrelevant information. However, its strong reliance on accurate labels makes it inherently vulnerable to label noise, prevalent in real-world scenarios, resulting in significant performance degradation and overfitting. To address this issue, we propose **LaT-IB**, a novel ***La**bel-Noise Resistan**T** **I**nformation **B**ottleneck* method which introduces a *"Minimal-Sufficient-Clean"* (MSC) criterion. Instantiated as a mutual information regularizer to retain task-relevant information while discarding noise, MSC addresses standard IB's vulnerability to noisy label supervision. To achieve this, LaT-IB employs a noise-aware latent disentanglement that decomposes the latent representation into components aligned with to the clean label space and the noise space. Theoretically, we first derive mutual information bounds for each component of our objective including prediction, compression, and disentanglement, and moreover prove that optimizing it encourages representations invariant to input noise and separates clean and noisy label information. Furthermore, we design a three-phase training framework: Warmup, Knowledge Injection and Robust Training, to progressively guide the model toward noise-resistant representations. Extensive experiments demonstrate that LaT-IB achieves superior robustness and efficiency under label noise, significantly enhancing robustness and applicability in real-world scenarios with label noise.

1 Introduction

The Information Bottleneck (IB) principle (Tishby, Pereira, and Bialek 2000) provides a fundamental theoretical framework for balancing compression and relevance in representation learning. Rooted in information theory, it has increasingly influenced the development of deep learning (Hu et al. 2024). IB encourages representations Z that retain only task-relevant information while discarding irrelevant or redundant input features using Mutual Information (MI) $I(\cdot;\cdot)$:

$$\min -I(Y;Z) + \beta I(X;Z). \tag{1}$$

IB-based methods aim to extract "Minimal-Sufficient" representations, inherently filtering out input noise and spurious correlations. This selective encoding mechanism contributes to their notable robustness under noisy or adversarial input perturbations (Shamir, Sabato, and Tishby 2010).

CIFAR10	40% asym noise	50% sym noise
ResNet34	77.78%	79.4%
VIB ($\beta = 0.01$)	73.80%	10.0%
Cora (40% noise)	Epoch: $0 \rightarrow 20$	Epoch: $20 \rightarrow 100$
GIB	22.9% $\rightarrow$ 69.5% steady increase $\uparrow$	69.5% $\rightarrow$ 55.1% steady decline $\downarrow$

Table 1: Performance of IB methods under noise conditions.

However, input noise rarely eliminates all useful information, allowing IB to extract meaningful features from Y. In contrast, label noise corrupts the supervisory signal, causing $I(Y;Z)$ to mislead Z to fit incorrect labels, thereby reducing robustness. This vulnerability is critical in real-world settings, where label noise is common and can severely harm performance, as real graphs are often disturbed by noise and unexpected factors. (Li et al. 2025). To address this, Label-Noise Representation Learning (LNRL) (Song et al. 2022) aims to extract robust features despite label corruption.

To empirically test the hypothesis that **IB is inherently vulnerable to label noise**, we conduct preliminary experiments on two tasks: image classification in computer vision and node classification in graph learning. We evaluate two representative IB-based methods: VIB (Alemi et al. 2017) and GIB (Wu et al. 2020). As shown in Table 1, VIB suffers performance drops and even training collapse, while GIB exhibits degraded accuracy. See Appendix E.3 for details.

To mitigate this, a simple remedy is to denoise the labels prior to applying IB. However, this two-stage pipeline is inherently suboptimal in both theory and practice.

Theorem 1.1 (Cumulative Degradation)**.** *In the two-stage approach, f_1 is used to modify the labels $Y' = f_1(\mathcal{D})$, and f_2 is responsible for extracting valid information from $\mathcal{D}$ while approximating the prediction result to $f_1(\mathcal{D})$. For one-stage model $g(\mathcal{D})$, it extracts the relevant information while removing noise. If the denoising abilities of f_1 and g are the same, the following inequality holds:*

$$P(f_2(\mathcal{D}) \neq g(\mathcal{D})) \geq \frac{H(Y'|\mathcal{D}) - 1}{\log(|\mathcal{Y}| - 1)}, \tag{2}$$

*Corresponding Author

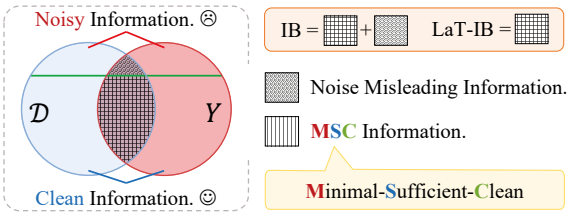

Figure 1: Comparison between LaT-IB and IB Principle.

where $\mathcal{Y}$ denotes the support of Y, and $|\mathcal{Y}|$ denotes the number of elements in $\mathcal{Y}$. The two models perform identically iff f_2 achieves the error lower bound and $H(Y'|\mathcal{D}) = 0$.

The proof of Theorem 1.1 is given in Appendix C.1. It demonstrates that cascading a denoising model f_1 with an IB learner f_2 leads to cumulative information loss compared with a unified model g, due to the extended information path. This phenomenon is further validated by empirical results, which show a clear degradation in the denoising effect when models are cascaded. See Appendix E.3 for detailed results.

Core Issue: *How can the IB principle be effectively applied to real-world scenarios with complex and unknown label noise, in order to learn representations that are* ***both "Minimal-Sufficient" and robust*** *to noisy supervision?*

Due to unknown label noise and the difficulty of integrating denoising with Information Bottleneck, applying IB in practice requires confronting the following key challenges:

- How to formulate the IB objective under label noise to learn clean representation. (▷ Section 4.1)
- How to optimize MI under label noise that distorts task-relevant representation learning. (▷ Section 4.2)
- How to effectively disentangle clean and noisy representations without knowing noisy samples. (▷ Section 4.3)

Present work. To address the core issue and tackle the key challenges, we propose a ***La**bel-Noise Resistan**T** **I**nformation **B**ottleneck* (**LaT-IB**) method. Centered on the idea of disentangling representations into clean and noisy label spaces, we formulate an IB training objective tailored for noisy supervision and theoretically justify its effectiveness through upper and lower bound analysis. To this end, we design a three-phase training framework: Warmup, Knowledge Injection and Robust Training, which gradually guides the model to learn "*Minimal-Sufficient-Clean*" (MSC) representations. A comparison between LaT-IB and standard IB principle is illustrated in Figure 1. Our contributions are:

- We identify the inherent vulnerability of IB to label noise and prove that denoising before IB is suboptimal.
- We propose a LaT-IB method that introduces MSC criterion of representations to enhance IB's robustness to label noise while maintaining its essential characteristics.
- We provide theoretical upper and lower bounds for LaT-IB, showing how disentangling clean and noise features enables robust representation learning. Based on this, we design a principled model and training framework.
- Extensive experiments evaluate LaT-IB's robustness and efficiency, outperforming baselines under label noise and adversarial attacks across diverse tasks and domains.

2 Related Work

2.1 Information Bottleneck for Robustness

The IB (Tishby, Pereira, and Bialek 2000) framework introduces a feature learning paradigm grounded in information theory. Works such as VIB (Alemi et al. 2017) and GIB (Wu et al. 2020) have advanced its practical use. Considering robustness, methods like DisenIB (Pan et al. 2021) and DGIB (Yuan et al. 2024) show reasonable robustness to input features, with studies (Xie et al. 2023; Pensia, Jog, and Loh 2020) further improving resilience to input noise.

Considering the presence of label noise, RGIB (Zhou et al. 2023) explores structural noise in GNNs to improve link prediction robustness. However, comprehensive studies on the vulnerability of IB to label noise still remain lacking.

2.2 Label-Noise Representation Learning

The LNRL aims to improve model robustness and representation quality under noisy label conditions. Existing approaches for learning with noisy labels include sample selection (Patel and Sastry 2023; Wei et al. 2020), which filters out likely noisy samples; robust loss functions (Zhang and Sabuncu 2018; Wang et al. 2019), which modify loss terms to reduce sensitivity to incorrect labels; noise-robust architectures (Liu et al. 2020), which use regularization to avoid overfitting noise; and data augmentation, such as mixup-based methods (Zhang et al. 2018; Harris et al. 2020), which interpolates samples to improve generalization.

However, most methods ignore representation-level constraints, making it hard to learn task-relevant and noise-invariant features under severe noise or distribution shifts.

3 Preliminary Analysis

Notation. We primarily define the input data $\mathcal{D}$. For vision tasks, $\mathcal{D} = X$, where $X \in \mathbb{R}^{N\times C\times P\times Q}$ denotes N samples with C channels and spatial size $P \times Q$ (e.g., height $\times$ width). For graph learning tasks, $\mathcal{D} = \mathcal{G} = (X, A)$, where $X \in \mathbb{R}^{N\times d}$ denotes d-dimensional features for N nodes and $A \in \mathbb{R}^{N\times N}$ represents the adjacency matrix. Each sample $\xi_i \in \mathcal{D}$ has a label $y_i \in Y$, which may be corrupted by noise during the labeling process. We denote Y_c and Y_n as the clean and noisy counterparts of Y respectively.

Analysis of IB Theory with Label Noise. In the traditional IB, $I(X;Z)$ encourages minimal representations by compressing the input, while $I(Y;Z)$ ensures sufficiency by preserving task-relevant information. However, when the label Y is corrupted by noise, maximizing $I(Y;Z)$ is equivalent to maximizing $I(Y_c, Y_n; Z)$, which inadvertently causes the learned representation Z to capture noise Y_n, thus compromising robustness and degrading performance.

In this study, we aim to mitigate the negative impact of label noise on model performance while preserving the "Minimal-Sufficient" property of the IB method. Ideally, we consider a robust IB method $\mathcal{M}_{IB}$ that, given a dataset $(\mathcal{D}, Y)$ where Y consists of both clean labels $y_i \in Y_c$ and noisy labels $y_j \in Y_n$, aims to satisfy the following objective:

$$\begin{aligned} &\min -I(Z;Y_c) + \beta I(Z;\mathcal{D}) \\ &\text{s.t. } Z = \mathcal{M}_{IB}(\mathcal{D}, Y). \end{aligned} \tag{3}$$

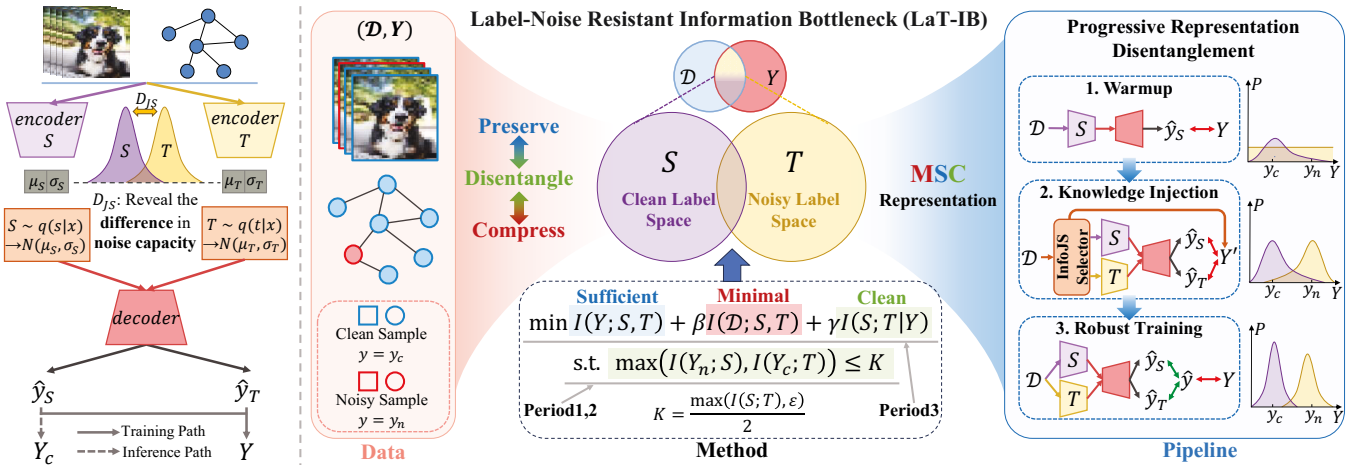

Figure 2: Left: The overall LaT-IB model architecture with dual encoders for extracting features from clean (S) and noisy (T) label spaces, and a shared decoder. Right: An illustration of the LaT-IB method, which disentangles representations to extract "Minimal-Sufficient-Clean" features. Specifically, its pipeline consists of three period: Warmup, Knowledge Injection and Robust Training, which transform Eq. (8) from a theoretical formulation into a practical training procedure.

Compared to the traditional IB objective, the goal of Eq. (3) is to maximize the MI between the learned representation and the clean labels Y_c , rather than with all observed labels Y. However, **whether each label is clean or noisy is unknown**. In the next section, we introduce a concrete solution to mitigate IB's vulnerability to label noise.

4 Methodology

In this paper, we propose Label-Noise Resistant Information Bottleneck (LaT-IB), along with theoretical formulation, model architecture and a tailored training framework, as illustrated in Figure 2. We begin by presenting the formal objective of LaT-IB and interpreting its theoretical implications. To enable efficient optimization, we derive upper and lower bounds that simplify the objective, effectively bridging the gap between theory and practice. Finally, drawing on key insights, we design a three-phase training framework: Warmup, Knowledge Injection and Robust Training, clarify the role of each phase and facilitate the progressive disentanglement of clean and noise-related representations.

4.1 Label-Noise Resistant Information Bottleneck

In real-world datasets, each training sample may have either a clean or a corrupted label, and sometimes both possibilities coexist probabilistically. Using a unified representation for all samples under such ambiguity can cause conflicting features and hurt downstream tasks. To mitigate this, we disentangle the representation into two parts: S under the clean label space, and T under the noise space. Under this disentanglement, the objective in Eq. (3) can be reformulated as:

$$\min -I(S;Y_c) + I(\mathcal{D};S,T). \tag{4}$$

Since only Y are available in the dataset, we implicitly associate it with the joint representation of S and T, where disentanglement is encouraged by $\min I(S;T|Y)$. A successful disentanglement implies that S and T encode conditionally independent given Y, capturing distinct semantics. With β and γ as balancing factors, the LaT-IB is formulated as:

$$\min \underbrace{-I(Y;S,T)}_{\text{prediction term}} + \beta \underbrace{I(\mathcal{D};S,T)}_{\text{compression term}} + \gamma \underbrace{I(S;T|Y)}_{\text{disentanglement term}}, \tag{5}$$

However, Eq. (5) still cannot map S to clean features and T to noise features. To address this and further explore its representational meaning, we introduce two lemmas below.

Lemma 4.1 (Nuisance Invariance)**.** *Taking the part of $\mathcal{D}$ that does not contribute to Y as $\mathcal{D}_n$ (D_n is independent of Y), and considering the Markov chain $(Y, \mathcal{D}_n) \to \mathcal{D} \to (S,T)$, the following inequality holds:*

$$I(\mathcal{D}_n;S,T) \le -I(Y;S,T) + I(\mathcal{D};S,T). \tag{6}$$

Lemma 4.2 (Feature Convergence)**.** *Assuming that Y can potentially contain all information about Y_c and Y_n, the following inequality holds when $\max(I(Y_n;S), I(Y_c;T)) \le \max(I(S;T),\varepsilon)/2 = K, \varepsilon > 0, \varepsilon \in \mathbb{R}$ is satisfied:*

$$-I(Y_c;S) - I(Y_n;T) - \varepsilon \le -I(Y;S,T) + I(S;T|Y). \tag{7}$$

The detailed proofs of these lemmas can be found in Appendix C.2. Lemma 4.1 demonstrates that optimizing $\min -I(Y;S,T) + I(\mathcal{D};S,T)$ in Eq. (5) ($\beta = 1$) essentially reduces the model's tendency to learn features irrelevant to Y (denoted as $\mathcal{D}_n$). Lemma 4.2 further indicates that, when the MI terms $I(Y_n,S)$ and $I(Y_c,T)$ are sufficiently small, optimizing $\min -I(Y;S,T) + I(S,T|Y)$ in Eq. (5) ($\gamma = 1$) effectively strengthens the mapping relationships $S \to Y_c$ and $T \to Y_n$. Based on these insights, we can first ensure the conditions in Lemma 4.2 then optimize the main objective in Eq. (5) as a form of **progressive representation disentanglement**. This enables the model to separate clean and noisy features while avoiding learning irrelevant noise $\mathcal{D}_n$.

By combining Lemma 4.1 and Lemma 4.2, we obtain a principled training objective that integrates sufficiency, com-

pression, and clean-noise disentanglement:

$$\begin{aligned}\min \underbrace{-I(Y;S,T)}_{\text{Sufficient}} + \beta \underbrace{I(\mathcal{D};S,T)}_{\text{Minimal}} + \gamma \underbrace{I(S;T|Y)}_{\text{Clean}} \\ \text{s.t. } \underbrace{\max(I(Y_n;S), I(Y_c;T)) \leq K}_{\text{Clean}}.\end{aligned} \tag{8}$$

4.2 Bound Analysis and Implementation

Building on the formulation introduced in the previous section, we now turn to the optimization of the proposed objective in Eq. (8). Since directly optimizing the multivariate MI is intractable, we first simplify the original objective by analyzing upper and lower bounds of MI, and then present the implementation strategy for each term. All proposition proofs are provided in the Appendix C.3.

Proposition 4.1 (The upper bound of $-I(Y;S,T)$)**.** *Given the label Y and the variable S,T that learns the characteristics of the clean label space and the noisy label space respectively, we have:*

$$-I(Y;S,T) \leq -\max\left(I(Y;S), I(Y;T)\right). \tag{9}$$

Intuitively, Eq. (9) encourages encoders to focus on learning its own knowledge, ensuring consistency in the learned representation. Further, since MI terms are intractable, each $I(Y,Z)$ with $Z \in \{S,T\}$ is lower-bounded by the cross-entropy loss using a variational approximation $q_\theta(y|z)$:

$$I(Y;Z) \geq \mathbb{E}_{p(y,z)}\left(\log(q_\theta(y|z))\right) := -\mathcal{L}_{CE}(Z,Y), \tag{10}$$

Proposition 4.2 (The upper bound of $I(\mathcal{D};S,T)$)**.** *Let $\mathcal{D}$, S, T be random variables. Assume the probabilistic mapping $p(\mathcal{D},S,T)$ follows the Markov chain $S \leftrightarrow \mathcal{D} \leftrightarrow T$. Then:*

$$I(\mathcal{D};S,T) \leq I(\mathcal{D};S) + I(\mathcal{D};T). \tag{11}$$

The implementation of each term $I(\mathcal{D};\cdot)$ remains consistent with that in VIB (Alemi et al. 2017) and GIB (Wu et al. 2020), achieved by minimizing the KL divergence between the variational posterior $q(\cdot|\mathcal{D})$ and the prior $p(\cdot)$.

Proposition 4.3 (Reformulation of $I(S,T|Y)$)**.** *Given the label Y and the variable S,T, minimizing $I(S;T|Y)$ is equivalent to minimize $I(S,Y;T,Y)$.*

The Proposition 4.3 achieves the tractable transformation of conditional MI theoretically. However, minimizing the term $I(S,Y;T,Y) = D_{\mathrm{KL}}\big[q(S,T,Y)||q(S,Y)q(T,Y)\big]$ is intractable since both distributions involve mixtures with many components. Therefore, we use the density-ratio trick (Sugiyama, Suzuki, and Kanamori 2012) by introducing a discriminator d, that learns to distinguish between samples from the joint distribution $q(s,t,y)$ and those from the product of marginals $q(s,y)q(t,y)$. In particular, we sample negative pairs $((s,y),(t,y'))$ from $q(s,y)q(t,y)$, where (s,y) and (t,y') are drawn independently, and positive pairs $((s,y),(t,y))$ from the joint distribution $q(s,t,y)$, where both s and t correspond to the same sample. The discriminator $d((s,y),(t,y'))$ is trained to output the probability that a given pair comes from the joint distribution, and the objective is to minimize the MI by solving the following problem:

$$\begin{aligned}\min_q \max_d \mathbb{E}_{q(s,y)q(t,y)} \log d((s,y),(t,y')) \\ + \mathbb{E}_{q(s,t,y)} \log(1 - d((s,y),(t,y))).\end{aligned} \tag{12}$$

When the discriminator cannot distinguish between joint and independent samples, the MI is effectively minimized.

Proposition 4.4 (Reformulation of the condition in Eq. (8): $\max(I(Y_n;S), I(Y_c;T)) \leq K$)**.** *Minimizing $I(Y_c;T)$ and $I(Y_n;S)$ is equivalent to maximize $I(Y_n;T)$ and $I(Y_c;S)$.*

Proposition 4.4 relaxes the condition in Eq. (8). Since the original MI calculation is mismatched and thus intractable, the relaxed formulation provides a tractable alternative that can be optimized efficiently, as described in Eq. (10).

4.3 Principle to Practice: LaT-IB Framework

Based on the theoretical analysis above, this section introduces the practical implementation of LaT-IB. To optimize the objective in Eq. (8), we adopt a three-phase training framework to **progressively disentangle the representation**. Specifically, we first introduce a **Warmup** period to provide the model with initial discriminative ability. Building on this, a **Knowledge Injection** period enforces the constraint by applying InfoJS selector, guiding the learning of encoder$_{S/T}$ via selected samples. Finally the **Robust Training** period focuses on optimizing the complete objective with prior knowledge, refining the model's robustness.

Feature-Decomposed Dual Encoder Architecture Design. Based on the Observation 4.1, we adopt the Jensen-Shannon (JS) divergence as a metric to evaluate the noise retention capacity of the two encoders.

Observation 4.1. *With the decoder kept fixed, we train the encoder using datasets that share the same input X but differ in the level of label noise in Y. As the noise gap between the two datasets increases, the divergence between the resulting encodings from the encoder also becomes larger.*

Accordingly, Figure 2 illustrates the overall architecture of the LaT-IB: the model is designed with a dual-encoder, single-decoder framework, where the two encoders extract features S and T, respectively. Each encoder maps the input features to a high-dimensional Gaussian distribution, and the embeddings are sampled using the reparameterization trick.

Phase 1: Warmup with Discriminative Learning under Noise. To address the problem of noise memorization during training, we introduce a Warmup phase where the model **builds basic discriminative ability**. Specifically, we pretrain the clean encoder encoder$_S$ using the full dataset, providing a foundation for more effective separation of clean and noisy samples in subsequent stages. The loss function in Warmup period is defined based on prediction $\hat{y}_S$:

$$\mathcal{L}_{Warmup} = \mathcal{L}_{CE}(\text{decoder}(S), Y) = \mathcal{L}_{CE}(\hat{y}_S, y). \tag{13}$$

Noise-Aware Sample Selection. Since the variables Y_c and Y_n are unobservable, we approximate the constraint $\max(I(Y_n;S), I(Y_c;T)) \leq K$ in Eq. (8) by selecting a partial set of confident samples to act as proxies for clean and noisy labels. Samples are then grouped into three categories for training: **Clean Set, Noise Set, and Uncertain Set**.

Observation 4.2. *For two different encoders, samples with more consistent predictions after passing through the decoder tend to have smaller divergence between their embeddings. In contrast, samples with inconsistent predictions correspond to larger embedding divergence.*

Observation 4.2 suggests that the divergence between encoders can be used to identify clean samples. Moreover, prior studies (Arpit et al. 2017; Song et al. 2019) have shown that models tend to fit clean samples earlier. Based on these insights, we designed the **InfoJS selector** as detailed in Algorithm B.2, which identifies clean (noisy) samples as those with MI between S and Y being in the top $\delta\%$ (bottom $\delta\%$) and JS divergence between S and T in the bottom $\delta\%$ (top $\delta\%$), respectively. Unselected samples are treated as the uncertain set. Labels are assigned as follows: $y' = y$ for Clean and Noise Sets, and $y' = g(\hat{y}_S)/g(\hat{y}_T)$ for Uncertain Set when training the encoder$_{S/T}$, where g denotes either a debiasing function (Menon et al. 2020) or one-hot mapping.

However, the InfoJS selector performs selection based on relative feature scores. To improve the quality of each set, we further enrich the sample composition by incorporating predicted confidence scores as an absolute criterion.

Phase 2: Knowledge Injection to Disentangle Representations. Once the model has acquired basic discriminative ability, we proceed to optimize the objective in Eq. (8). Given the condition and its reformulated form:

$$\begin{aligned} &\underbrace{\max(I(Y_n;S), I(Y_c;T)) \leq K}_{\text{The original constraint in Eq. (8)}} \\ \Rightarrow\ &\underbrace{\max(I(Y_n;T)), \max(I(Y_c;S))}_{\text{The reformulated constraint in Proposition 4.4}}, \end{aligned} \tag{14}$$

to satisfy the constraint, we introduce a Knowledge Injection phase to encourage the encoder$_{S,T}$ to learn disentangled representations. Furthermore, to enforce difference in noise representation between the two encoders, we incorporate the JS divergence D_{JS} based on Observation 4.1:

$$\begin{cases} \mathcal{L}_{Clean} = \mathcal{L}_{CE}(\hat{y}_S, y') - D_{JS}(s \parallel t), \\ \mathcal{L}_{Uncertain} = \mathcal{L}_{CE}(\hat{y}_S, y') + \mathcal{L}_{CE}(\hat{y}_T, y') + D_{JS}(s \parallel t), \\ \mathcal{L}_{Noise} = \mathcal{L}_{CE}(\hat{y}_T, y') - D_{JS}(s \parallel t), \end{cases} \tag{15}$$

For Clean and Noise Sets, divergence is maximized to increase encoder discrepancy; and for the Uncertain set, divergence is minimized to guide T towards meaningful patterns. It is worth noting that the Uncertain set is much smaller, thus has limited influence on the encoders' training process.

Empirically, minimizing the $I(\mathcal{D}; S, T)$ helps to leading to a more robust encoding space. To progressively disentangle the representation and achieve a minimal representation, we introduce a regularization term $\mathcal{L}_{Minimal}$ that approximates the $\min I(\mathcal{D}; S, T)$ term base on Proposition 4.2, and incorporate it into the loss function during the Knowledge Injection period to learn a compact representation:

$$\mathcal{L}_{Injection} = \mathcal{L}_{Clean} + \mathcal{L}_{Uncertain} + \mathcal{L}_{Noise} + \mathcal{L}_{Minimal}. \tag{16}$$

This facilitates a smoother transition to the third Robust Training stage. The implementation details of $\mathcal{L}_{Minimal}$ are provided in the Appendix D.1.

Phase 3: Robust Training for Representation Consistency. The Warmup stage establishes initial discriminative ability, while Knowledge Injection realized constraint to guide the model toward informative and reliable samples. To further disentangle and enhance representation robustness under label noise, this period focuses on optimizing the full objective in Eq. (5): $\min -I(Y; S, T) + I(\mathcal{D}; S, T) + I(S; T|Y)$, aiming to learn noise consistent representations.

Section 4.2 has introduced the implementation of each objective term. Among them we propose $\mathcal{L}_{ConCE}$ to optimize the term $I(Y; S, T)$ based Eq. (9) and (10):

$$\mathcal{L}_{ConCE} \leftarrow \sum \min(\mathcal{L}_{CE}(\hat{y}_S, y), \mathcal{L}_{CE}(\hat{y}_T, y)), \tag{17}$$

encouraging consistency between encoders and clean/noisy labels. Detailed formulations are provided in Appendix B.1.

The loss function for the Robust Training period is:

$$\begin{aligned} \mathcal{L}_{Robust} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{\mathcal{B}} [&\underbrace{\mathcal{L}_{ConCE}(\hat{y}_S, \hat{y}_T, y)}_{\text{Eq. (17)}} + \beta \underbrace{\mathcal{L}_{Minimal}}_{\text{Eq. (11)}} \\ &- \gamma \underbrace{\log d(s_i, y_i; t_i, y_i)}_{\text{Proposition 4.3, Eq. (12)}}], \end{aligned} \tag{18}$$

where $\mathcal{B}$ denotes a training batch. In addition, we alternately update the discriminator d based on Eq. (12), using a random permutation π to approximate the marginal distribution:

$$\begin{aligned} \mathcal{L}_d = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{\mathcal{B}} &- \log(1 - d(s_i, y_i; t_{\pi(i)}, y_{\pi(i)})) \\ &- \log d(s_i, y_i; t_i, y_i). \end{aligned} \tag{19}$$

5 Experiment

In this section, we conduct extensive experiments to evaluate the robustness and efficiency of the LaT-IB under diverse tasks and various types of noise, including real-world and synthetic label noise, as well as adversarial perturbation. [1]

5.1 Experimental Settings

Datasets. We evaluate the proposed LaT-IB method on multiple datasets. For image classification, we utilize the CIFARN (Wei et al. 2022), Animal-10N (Song, Kim, and Lee 2019) and CIFAR (Krizhevsky, Hinton et al. 2009) datasets. For node classification tasks, we evaluate on Cora, Citeseer, Pubmed (Sen et al. 2008), and DBLP (Pan et al. 2016). More descriptions about datasets are provided in Appendix E.1.

Baselines. We compare our LaT-IB with four categories, 16 baselines in two scenarios: ① Classic IB methods; ② IB with robust loss functions; ③ Improved IB variants; ④ Two-stage denoising + IB methods. They comprehensively evaluate our LaT-IB's performance from multiple perspectives.

Label Noise Settings. To evaluate the robustness of LaT-IB and baselines against label noise, we conduct experiments in both image and graph classification tasks. For image classification, we evaluate on both real-world noisy datasets and synthetic settings with symmetric and asymmetric label noise, simulated using custom transition matrices as described in (Xiao et al. 2023). For node classification, we follow the protocol in (Wang et al. 2024) to inject uniform and pairwise label noise into graph labels.

[1] Code available at: https://github.com/RingBDStack/LaT-IB

Method	Model	CIFAR-10N					CIFAR-100N	Animal -10N
		aggre	rand1	rand2	rand3	worst	noisy100	
Classic	VIB	$86.11_{\pm 0.34}$	$83.69_{\pm 0.50}$	$83.69_{\pm 0.46}$	$83.76_{\pm 0.29}$	$73.80_{\pm 0.59}$	$53.29_{\pm 0.09}$	$76.28_{\pm 0.51}$
IB	NIB	$85.21_{\pm 0.44}$	$84.03_{\pm 1.43}$	$81.98_{\pm 0.68}$	$82.39_{\pm 0.43}$	$73.51_{\pm 0.82}$	$48.11_{\pm 0.40}$	$75.62_{\pm 0.64}$
Robust	VIB ($\mathcal{L}_{GCE}$)	$85.70_{\pm 0.08}$	$84.32_{\pm 0.50}$	$83.97_{\pm 0.38}$	$84.25_{\pm 0.68}$	$78.88_{\pm 0.27}$	—	$81.72_{\pm 1.77}$
Loss	VIB ($\mathcal{L}_{SCE}$)	$83.95_{\pm 0.10}$	$82.65_{\pm 0.25}$	$82.84_{\pm 0.31}$	$82.50_{\pm 0.24}$	$73.81_{\pm 1.54}$	$50.71_{\pm 0.14}$	$77.17_{\pm 0.44}$
Improved	SIB	$89.99_{\pm 0.08}$	$84.75_{\pm 1.04}$	$85.07_{\pm 0.72}$	$85.39_{\pm 0.50}$	$70.58_{\pm 0.50}$	$50.82_{\pm 0.41}$	$83.95_{\pm 0.14}$
IB	DT-JSCC	$85.46_{\pm 0.44}$	$81.85_{\pm 0.66}$	$81.14_{\pm 0.55}$	$81.03_{\pm 0.34}$	$69.73_{\pm 1.15}$	$43.61_{\pm 0.19}$	$78.98_{\pm 0.23}$
Deniose	JoCoR+VIB	$86.39_{\pm 0.18}$	$86.45_{\pm 0.02}$	$86.53_{\pm 0.29}$	$86.60_{\pm 0.11}$	$81.65_{\pm 0.15}$	$54.24_{\pm 0.18}$	$75.45_{\pm 0.27}$
+ IB	(ELR+)+VIB	$\underline{92.65_{\pm 0.27}}$	$92.09_{\pm 0.25}$	$92.01_{\pm 0.20}$	$91.93_{\pm 0.15}$	$86.68_{\pm 0.25}$	$61.06_{\pm 0.34}$	$\underline{85.87_{\pm 0.15}}$
	Promix+VIB	$92.35_{\pm 0.38}$	$\underline{92.59_{\pm 0.40}}$	$\underline{92.42_{\pm 0.17}}$	$\underline{92.54_{\pm 0.21}}$	$\mathbf{91.24_{\pm 0.28}}$	$\mathbf{63.91_{\pm 0.19}}$	$85.47_{\pm 0.51}$
Ours	LaT-IB	$\mathbf{94.17_{\pm 0.12}}$	$\mathbf{93.25_{\pm 0.11}}$	$\mathbf{93.19_{\pm 0.09}}$	$\mathbf{93.03_{\pm 0.11}}$	$\underline{87.95_{\pm 0.22}}$	$\underline{63.59_{\pm 0.67}}$	$\mathbf{88.49_{\pm 0.11}}$

Table 2: Classification accuracy (%) on the CIFAR-10N/100N and Animal-10N dataset. All the best results are highlighted in **bold**, and the second-best results are underlined.

Method	Model	Clean	Uniform Noise				Pair Noise			
			10%	20%	30%	40%	10%	20%	30%	40%
Classic	GIB	$71.57_{\pm 1.18}$	$\underline{70.50_{\pm 1.85}}$	$64.30_{\pm 6.45}$	$63.90_{\pm 3.51}$	$\underline{62.67_{\pm 1.35}}$	$68.67_{\pm 3.47}$	$61.30_{\pm 14.57}$	$67.53_{\pm 4.77}$	$55.57_{\pm 14.33}$
Robust	GIB ($\mathcal{L}_{GCE}$)	$69.93_{\pm 0.69}$	$67.43_{\pm 3.21}$	$61.67_{\pm 7.19}$	$47.80_{\pm 18.62}$	$43.47_{\pm 14.50}$	$50.93_{\pm 0.52}$	$55.33_{\pm 11.23}$	$62.37_{\pm 6.99}$	$36.90_{\pm 15.23}$
Loss	GIB ($\mathcal{L}_{SCE}$)	$\underline{72.53_{\pm 0.12}}$	$70.17_{\pm 2.10}$	$\underline{71.63_{\pm 2.05}}$	$62.90_{\pm 8.09}$	$51.87_{\pm 6.03}$	$69.30_{\pm 1.66}$	$68.23_{\pm 3.41}$	$65.13_{\pm 5.02}$	$51.13_{\pm 11.30}$
Improv-	CurvGIB	$64.63_{\pm 5.28}$	$65.67_{\pm 5.85}$	$54.67_{\pm 10.09}$	$54.00_{\pm 2.41}$	$54.97_{\pm 2.78}$	$59.97_{\pm 9.00}$	$62.07_{\pm 5.15}$	$66.63_{\pm 1.94}$	$54.57_{\pm 1.25}$
ed IB	IS-GIB	$71.00_{\pm 1.22}$	$69.97_{\pm 1.41}$	$64.30_{\pm 2.30}$	$59.77_{\pm 3.70}$	$53.77_{\pm 4.41}$	$64.83_{\pm 2.34}$	$62.50_{\pm 1.51}$	$62.50_{\pm 1.31}$	$55.40_{\pm 4.74}$
Denoise	RNCGLN+GIB	$70.57_{\pm 0.99}$	$69.50_{\pm 0.86}$	$63.43_{\pm 5.90}$	$62.83_{\pm 4.15}$	$53.27_{\pm 14.47}$	$\underline{69.90_{\pm 1.69}}$	$68.20_{\pm 2.14}$	$66.47_{\pm 3.21}$	$\underline{56.77_{\pm 15.11}}$
+ IB	CGNN+GIB	$71.87_{\pm 1.99}$	$68.97_{\pm 3.09}$	$65.47_{\pm 4.77}$	$\underline{64.93_{\pm 2.46}}$	$48.83_{\pm 6.48}$	$59.03_{\pm 12.67}$	$\underline{69.77_{\pm 1.77}}$	$\underline{68.50_{\pm 2.83}}$	$53.93_{\pm 13.85}$
Ours	LaT-IB	$\mathbf{74.97_{\pm 0.68}}$	$\mathbf{74.90_{\pm 2.09}}$	$\mathbf{73.40_{\pm 2.62}}$	$\mathbf{70.50_{\pm 3.86}}$	$\mathbf{72.20_{\pm 4.22}}$	$\mathbf{75.63_{\pm 0.46}}$	$\mathbf{73.03_{\pm 1.77}}$	$\mathbf{70.07_{\pm 3.20}}$	$\mathbf{68.77_{\pm 2.29}}$

Table 3: Classification accuracy (%) on the Pubmed dataset under different noise types and noise rates. All the best results are highlighted in **bold**, and the second-best results are underlined.

Adversarial Attack Settings. As discussed in Appendix D.1, the implementation of $I(\mathcal{D}; S, T)$ in our LaT-IB framework aligns with prior work VIB and GIB, thereby theoretically inheriting their robustness properties. To empirically verify this claim, we evaluate LaT-IB's performance under adversarial perturbations in the image classification setting. Specifically, we adopt the FGSM (Goodfellow, Shlens, and Szegedy 2015) attack to perturb input images with $\varepsilon \in \{0.05, 0.1, 0.2\}$, controlling the perturbation strength. Combined with noisy labels during training, this setup evaluate the robustness of model under compound noise conditions.

5.2 Robustness Against Label Noise

In this section, we evaluate the representation capability of our proposed method under various label noise conditions. Specifically, we investigate whether the LaT-IB model can effectively learn robust representations when trained on data corrupted by different types and levels of label noise.

Results. In most scenarios, our proposed LaT-IB method outperforms other baseline approaches as shown in Table 2 and 3. In certain cases, however, methods that first perform denoising and then apply IB achieve better results, likely due to the strong denoising capacity of those models. Nevertheless, such two-stage methods involve longer training pipelines and are more vulnerable to adversarial attacks, as will be demonstrated in the next section. Additional experimental results on label noise are shown in Appendix E.

5.3 Robustness Against Adversarial Perturbations

In this section, to further validate the "Minimal-Sufficient" property in MSC of the proposed LaT-IB method, we apply perturbations to the input data $\mathcal{D}$. The perturbed data is then fed into models trained under noisy label settings. This setup enables a comprehensive evaluation of model robustness against diverse noise, including inputs and labels.

Results. The results demonstrate that the LaT-IB method exhibits strong robustness against adversarial attacks, significantly outperforming other approaches, as shown in Table 4. Notably, two-stage methods suffer a substantial performance drop under attack due to the increased number of vulnerable components, further highlighting their limitations.

5.4 Ablation Study

In this section, we analyze the effectiveness of different training stages in the LaT-IB framework. To investigate the

Model	CIFAR-10N (aggre)				CIFAR-10N (worst)			
	No attack	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	No attack	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
VIB	$86.11_{\pm 0.34}$	$52.33_{\pm 1.55}$	$43.18_{\pm 2.10}$	$36.63_{\pm 1.10}$	$73.80_{\pm 0.59}$	$43.29_{\pm 2.36}$	$\underline{36.56}_{\pm 3.28}$	$32.17_{\pm 3.27}$
VIB ($\mathcal{L}_{GCE}$)	$85.70_{\pm 0.08}$	$54.15_{\pm 1.85}$	$44.84_{\pm 3.00}$	$34.72_{\pm 2.43}$	$78.88_{\pm 0.27}$	$43.27_{\pm 1.56}$	$31.24_{\pm 1.42}$	$24.23_{\pm 1.45}$
SIB	$89.99_{\pm 0.08}$	$\underline{56.48}_{\pm 2.50}$	$\underline{46.62}_{\pm 2.41}$	$\underline{38.14}_{\pm 2.37}$	$70.58_{\pm 0.50}$	$43.39_{\pm 2.83}$	$33.40_{\pm 2.87}$	$\underline{27.89}_{\pm 2.93}$
(ELR+)+VIB	$\underline{92.65}_{\pm 0.27}$	$39.88_{\pm 0.74}$	$23.16_{\pm 0.40}$	$14.60_{\pm 0.39}$	$86.68_{\pm 0.25}$	$42.72_{\pm 0.24}$	$26.44_{\pm 0.64}$	$14.70_{\pm 0.70}$
Promix+VIB	$92.35_{\pm 0.38}$	$51.27_{\pm 1.53}$	$36.43_{\pm 0.65}$	$20.49_{\pm 1.79}$	$\mathbf{91.24}_{\pm 0.28}$	$\underline{52.88}_{\pm 1.46}$	$36.05_{\pm 0.68}$	$23.79_{\pm 0.14}$
LaT-IB	$\mathbf{94.17}_{\pm 0.12}$	$\mathbf{69.38}_{\pm 1.23}$	$\mathbf{60.66}_{\pm 2.03}$	$\mathbf{49.64}_{\pm 2.27}$	$\underline{87.95}_{\pm 0.22}$	$\mathbf{64.36}_{\pm 1.67}$	$\mathbf{54.18}_{\pm 2.53}$	$\mathbf{43.91}_{\pm 3.05}$

Table 4: Classification accuracy (%) on CIFAR-10N (aggre and worst) under different adversarial perturbation levels. For Denoise + IB methods, adversarial attacks are applied in both stages: the VIB model is trained using the output of a denoising model that has itself been attacked. All the best results are highlighted in **bold**, and the second-best results are underlined.

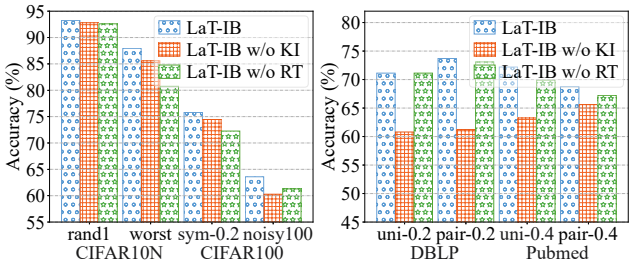

Figure 3: Ablation study.

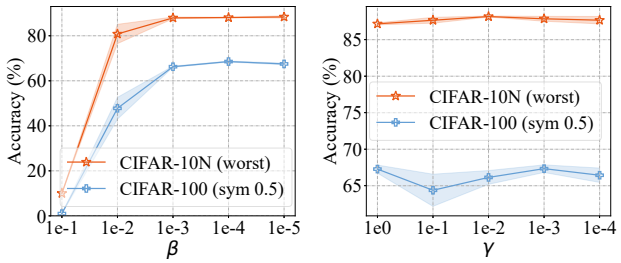

Figure 4: The influence of β and γ.

role of each phase in enhancing model robustness, we design two ablated variants:

- **LaT-IB (*w/o KI*)**: We remove the Knowledge Injection period, thus $\max(I(Y_n;S), I(Y_c;T)) \leq K$ is not satisfied. weakening the ability to map $S \to Y_c$ and $T \to Y_n$.
- **LaT-IB (*w/o RT*)**: We remove the Robust Training period, meaning no further enhancement is applied to the representations from S, T. The LaT-IB model can only gain partial information from the three subsets.

Note that we do not design an ablation variant without the Warmup period, as it is essential for establishing basic classification capability and stable later training.

Results. Overall, the full LaT-IB method achieves the best performance under all noisy label settings as shown in Figure 3, demonstrating the importance of different periods in the framework. For image classification tasks (with larger samples), the Robust Training stage is particularly critical, while for graph-based tasks (with fewer samples), the Knowledge Injection stage proves more influential. These findings highlight the necessity of each training stage in achieving robust representations under noisy supervision.

5.5 Hyperparameter Sensitivity Analysis

We analyze the sensitivity of the model to the hyperparameter β, γ and δ. The coefficient β controls the feature compression term $I(\mathcal{D};S,T)$, which encourages the model to learn noise invariant features. The coefficient γ controls the feature separation term $I(S;T|Y)$, which encourages the encoder$_{S,T}$ to capture clean and noisy representations respectively. δ regulates how much information the encoder$_{S,T}$ learns during the Knowledge Injection phase.

Results. We observe that a large β can dominate training and cause collapse as shown in Figure 4, indicating that $I(\mathcal{D};S,T)$ partially limits the model's expressiveness. However, our method is more tolerant to β than the original VIB, which fails to train on CIFAR-10 with 50% symmetric noise at $\beta = 0.01$. In contrast, our model performs better as β decreases because the input compression level is reduced.

We also observe that the model's sensitivity to γ varies across noisy settings, highlighting the importance of the separation term $I(S;T|Y)$ under different types of noise.

For δ, a too-small δ limits the encoder's training data exposure, while a too-large δ causes the encoders to converge, reducing their ability to separate clean and noisy information. More detailed results in Appendix E.6.

6 Conclusion

In this work, we propose **LaT-IB**, a novel yet principled IB framework that enables robust representation learning under label noise while preserving the principle of learning minimally sufficient representations. We disentangle features into representations related to clean and noisy label spaces, and theoretically demonstrate the noise-separating effect of our method through upper and lower bounds analysis. Furthermore, we design a three-phase training framework comprising Warmup, Knowledge Injection and Robust Training, that facilitates the extraction of *"Minimal-Sufficient-Clean"* representations. Extensive experiments across diverse noisy environments validate the superior performance of LaT-IB compared to existing IB-based methods, highlighting its potential to efficiently advance the practical application of IB theory in real-world learning scenarios with label noise.

Acknowledgments

The corresponding author is Qingyun Sun. This work is supported by NSFC under grants No.62427808 and No.62225202, and by the Fundamental Research Funds for the Central Universities. We extend our sincere thanks to all reviewers for their valuable efforts.

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *ICLR*.

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *ICML*, 233–242.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

Harris, E.; Marcu, A.; Painter, M.; Niranjan, M.; Prügel-Bennett, A.; and Hare, J. 2020. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*.

Hu, S.; Lou, Z.; Yan, X.; and Ye, Y. 2024. A survey on information bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, B.; Xie, X.; Lei, H.; Fang, R.; and Kang, Z. 2025. Simplified PCNet with robustness. *Neural Networks*, 184: 107099.

Liu, S.; Niles-Weed, J.; Razavian, N.; and Fernandez-Granda, C. 2020. Early-learning regularization prevents memorization of noisy labels. *NeurIPS*, 33: 20331–20342.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*.

Pan, S.; Wu, J.; Zhu, X.; Zhang, C.; and Wang, Y. 2016. Tri-party deep network representation. In *IJCAI*, 1895–1901.

Pan, Z.; Niu, L.; Zhang, J.; and Zhang, L. 2021. Disentangled information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9285–9293.

Patel, D.; and Sastry, P. 2023. Adaptive sample selection for robust learning under label noise. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3932–3942.

Pensia, A.; Jog, V.; and Loh, P.-L. 2020. Extracting robust and accurate features via a robust information bottleneck. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 131–144.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shamir, O.; Sabato, S.; and Tishby, N. 2010. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30): 2696–2711.

Song, H.; Kim, M.; and Lee, J.-G. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 5907–5915.

Song, H.; Kim, M.; Park, D.; and Lee, J.-G. 2019. How does early stopping help generalization against label noise? *arXiv preprint arXiv:1911.08059*.

Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11): 8135–8153.

Sugiyama, M.; Suzuki, T.; and Kanamori, T. 2012. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64: 1009–1044.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, 322–330.

Wang, Z.; Sun, D.; Zhou, S.; Wang, H.; Fan, J.; Huang, L.; and Bu, J. 2024. NoisyGL: A Comprehensive Benchmark for Graph Neural Networks under Label Noise. *arXiv preprint arXiv:2406.04299*.

Wei, H.; Feng, L.; Chen, X.; and An, B. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13726–13735.

Wei, J.; Zhu, Z.; Cheng, H.; Liu, T.; Niu, G.; and Liu, Y. 2022. Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations. In *ICLR*.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph information bottleneck. *NeurIPS*, 33: 20437–20448.

Xiao, R.; Dong, Y.; Wang, H.; Feng, L.; Wu, R.; Chen, G.; and Zhao, J. 2023. ProMix: combating label noise via maximizing clean sample utility. In *Proceedings of the Thirty-Second IJCAI*, 4442–4450.

Xie, S.; Ma, S.; Ding, M.; Shi, Y.; Tang, M.; and Wu, Y. 2023. Robust information bottleneck for task-oriented communication with digital modulation. *IEEE Journal on Selected Areas in Communications*, 41(8): 2577–2591.

Yuan, H.; Sun, Q.; Fu, X.; Ji, C.; and Li, J. 2024. Dynamic graph information bottleneck. In *Proceedings of the ACM Web Conference 2024*, 469–480.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *ICLR*.

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *NeurIPS*, 31.

Zhou, Z.; Yao, J.; Liu, J.; Guo, X.; Yao, Q.; He, L.; Wang, L.; Zheng, B.; and Han, B. 2023. Combating bilateral edge noise for robust link prediction. *NeurIPS*, 36: 21368–21414.