EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Scalable Machine Learning Generalization Analysis Model for Intelligent Information Systems Incorporating PAC-Bayes Bounds

Lina Guo[1], Zijing Zhang[2,*], Yujia Zhang[3] and Le Ji[3]
[1] School of General Education, Shandong Huayu University of Technology, Dezhou 253034, Shandong, China
[2] School of Economics and Management, Shandong Huayu University of Technology, Dezhou 253034, Shandong, China
[3] School of Innovation and Entrepreneurship, Shandong Huayu University of Technology, Dezhou 253034, Shandong, China

Abstract

This study proposes SGAM-PB, a scalable machine-learning generalization analysis model that incorporates PAC-Bayes bounds. The experimental results show that the SGAM-PB achieves the best overall performance across five intelligent information system tasks. Compared with ERM, the generalization gap is reduced from 0.118 to 0.036, a decrease of approximately 69.5%. The PAC-Bayes bound is reduced to approximately 0.15 after 100 training epochs, showing a smoother and tighter convergence trend than the baseline methods. Under 50% drift intensity, the SGAM-PB maintains an accuracy of approximately 0.77, outperforming the ERM by nearly 0.10. In dynamic knowledge update experiments, the recovered accuracy reached approximately 0.86 after 30 update steps, which was higher than that of the ERM and traditional PAC-Bayes baseline. In scalability testing, SGAM-PB maintains the runtime cost below that of the deep ensemble and BNN methods when the data volume increases to 1000 K samples. These results demonstrate that the SGAM-PB can effectively reduce deployment risk, improve drift robustness, accelerate update recovery, and provide interpretable generalization monitoring for intelligent information systems.

Keywords: PAC-Bayes bound, machine learning generalization ability, intelligent information system, knowledge management, data drift, knowledge update, model reliability, scalable learning

Received on 25 Feburary 2026, accepted on 18 April 2026, published on 16 September 2026

doi: 10.4108/eetsis.14496

*Corresponding author. Email: 17852349859@163.com

1. Introduction

Intelligent information systems have evolved from data storage, retrieval, and transmission platforms to knowledge service systems driven by machine learning models. For example, enterprise knowledge management, digital libraries, intelligent retrieval, online learning systems, scientific research knowledge organization, recommendation and decision support systems, and machine learning-based models can be used for knowledge classification, semantic matching, entity recognition, user profiling, knowledge recommendation, anomaly detection, and risk identification tasks. These models can extract patterns from massive heterogeneous data, improve information processing efficiency, and facilitate the personalization and intelligence of knowledge-based services. Consequently, model reliability under deployment and drift is a practical concern for knowledge-service systems, as demonstrated in deployed learning analytics.

ML models on smart information systems do not function in static environments. In particular, the data change every day in real information systems: knowledge bases are updated, documents, concepts, entities, topics, and relationships are added to the system, and old knowledge is removed, deleted, or annotated; user behaviour is changed in terms of organization, business

context, time, and external events. Users search, click behaviour, browsing, favourites, and feedback may all be lost; domain terminology is dynamically evolving; some keywords can have different semantics at different times; and new abbreviations, professional expressions, and cross-domain concepts are appearing in the literature. Finally, as the system increases, label noise, feedback sparsity, sample imbalance, and missing data come into play, which results in variations in the training data distribution as well as the deployed data distribution, thus resulting in degraded predictive performance and increased generalization risk under distribution shift.

Generalization ability. The generalization ability must be able to guarantee the performance of a machine learning model on unseen data. Generalization analysis normally considers the gap between the training and test errors; however, in intelligent information systems, the generalization problem is more challenging, where the system must deal with the gap between training and testing sets, with knowledge drift, user drift, semantic drift, and label drift over time. Therefore, intelligent systems require an analytic approach that provides empirical risk, model complexity, distribution drift, upper bound of the true risk, and risk of future model failure. This will be used for retraining, updating, and knowledge management.

The PAC-Bayes theory provides the key theoretical context for such problems. PAC-Bayes combines the concept of probabilistic approximate correct learning with Bayesian learning, characterizing model complexity through KL divergence between prior and posterior distributions and constructing an upper bound on the true risk with empirical risk. Instead of simply minimizing the empirical risk, the PAC-Bayes analysis is weighted between the empirical error and model complexity. If a model performs well on the training data but the posterior distribution is far from the prior, its generalization bound may not be tight. This complexity penalty can make the guarantee loose even when the training error is low.

PAC-Bayesian research has addressed practical disintegration and neural-network generalization [1-3], multi-view learning [4], and lifelong multi-armed bandits [5]. Separately, drift studies cover pipeline mitigation [6], performance-aware detection [7], domain-adaptation-based handling [8], locality and recovery [9], localization and explanation [10], continuous adaptation [11], recurring drift [12], and active/passive handling [13]. These strands do not yet provide a unified, scalable treatment of knowledge-base updates, user-behaviour drift, and semantic migration in intelligent information systems. Recent PAC-Bayesian research has extended generalization analysis beyond fixed-sample and i.i.d. settings. Time-uniform bounds remain valid during sequential monitoring and at data-dependent stopping times, including settings with non-stationary losses and non-i.i.d. observations [14,15]. PAC-Bayesian analyses of data-dependent hypothesis sets and transductive learning further cover adaptive algorithms, semi-supervised learning, and graph learning [16,17]. Instance-dependent generalization bounds provide complementary guarantees under distribution shifts [18], while scalable PAC-Bayesian meta-learning shows how prior information can be transferred across related tasks with explicit complexity control [19]. Building on these advances, the present study integrates PAC-Bayes generalization analysis, drift-aware calibration, and scalable posterior updating into a unified risk-monitoring architecture for continuously updated intelligent information systems.

Against this background, we developed SGAM-PB, a scalable intelligent information system machine learning generalization capability analysis model based on PAC-Bayes bounds. This model addresses the issue of the deployment risk of machine learning models in dynamic knowledge environments through a data acquisition layer, feature representation layer, prior-posterior learning layer, PAC-Bayes bound estimation layer, drift-aware calibration layer, incremental KL update layer, and knowledge risk output layer. The SGAM is based on both PAC-bayes bounds, model complexity in KL divergence, knowledge environment changes in drift intensity, and an interpretable generalization risk score via a risk function.

The main contributions of this study are as follows: First, a framework for analysing the generalization ability of machine learning for intelligent information systems is proposed, extending the PAC-Bayes bound from traditional theoretical learning scenarios to knowledge management and dynamic information service scenarios. Second, a generalization risk scoring function is constructed that integrates empirical risk, PAC-Bayes bound, model complexity, and knowledge drift intensity, expanding the model reliability evaluation from a single accuracy comparison to a multi-dimensional risk interpretation. Third, an incremental KL divergence update mechanism was designed to reduce the computational cost of repeatedly estimating prior-posterior complexity in large-scale knowledge systems. Fourth, a deep experimental simulation system is constructed that includes knowledge document classification, user behaviour prediction, knowledge recommendation, dynamic knowledge update, and cross-domain drift, verifying the model's effectiveness from the perspectives of generalization gap, bound tightness, drift robustness, recovery ability, and scalability. Fifth, a real-world visualization analysis of knowledge drift under different algorithm processing methods is designed to demonstrate how the model identifies risk nodes, tracks knowledge drift paths, and explains the impact on downstream tasks.

2. Related Work

2.1. Generalization Analysis for Dynamic Intelligent Information Systems

The goal of intelligent information systems is to learn, organize, share, and use information through information processing. With the recent development of machine learning, intelligent information has shifted away from

rule- and manual configuration to data- and model-driven information. Knowledge document classification. In a knowledge document classification problem, ML models can automatically select document categories based on the text semantics, metadata, and topic features. In semantic retrieval problems, vector representation models can calculate the semantic similarity between queries and documents to increase the relevance of the retrieval results. In knowledge recommendation tasks, ML models can predict the type of knowledge resource users might need by exploiting user behaviour, knowledge content, and interactions. In risk identification problems, ML-based models detect abnormal logs, inaccurate knowledge links, expired documents, and inconsistent labels.

From a learning-theoretic perspective, intelligent information systems can be viewed as collections of related prediction tasks that evolve as documents, users, entities, and interaction patterns change. Their reliability therefore depends not only on task accuracy but also on whether risk guarantees remain valid during sequential updates, whether the learned hypothesis set depends on observed data, and whether prior knowledge can be transferred across related tasks without excessive complexity growth. Recent PAC-Bayesian studies address these issues through time-uniform bounds [14,15], bounds for data-dependent hypothesis sets and transductive learning [16,17], distribution-shift-sensitive generalization analysis [18], and scalable PAC-Bayesian meta-learning [19]. These results motivate evaluating intelligent information systems through empirical risk, posterior complexity, update dynamics, and deployment shift in a unified framework.

2.2. Machine Learning Generalization Risk and Model Reliability

Generalization ability is the ability of a machine learning model to maintain good performance on unseen samples. Typically, ML estimates the generalization ability by dividing the data into training, validation, and testing sets. If the test set has the same distribution as the real deployment data, the test performance can be assumed to be approximately the deployment effect of a model. This assumption does not hold true for intelligent information systems. Knowledge objects, user behaviours, and system tasks vary over time; therefore, the test does not have a clear view of the future.

The generalization risk refers to empirical risk, model complexity, sample size, training strategy, and data distribution. A lower empirical risk does not necessarily lead to a lower real risk because the model can potentially overfit the training samples. Higher model complexity is generally more expressive but may also result in model overfitting. Small sample sizes are often better for generalization errors, but large-scale historical data may mislead the model if the data distribution is changed. For smart information systems, model reliability should be assessed from a long-term operational perspective, rather than from a single offline test.

Model reliability is also an important task in deployment environments and may provide uncertainty estimates, drift detection, and risk warnings. When the model has high confidence in making a wrong prediction, it might even send incorrect guesses to the user. If the model does not know when changes in the input distribution have occurred, it can only notice once the performance has failed. Deployment reliability requires monitoring not only predictive accuracy but also the mechanisms by which performance changes under distribution shift. Performance-aware drift detectors connect concept drift to model degradation [7], while domain-adaptation analyses provide explicit generalization guarantees for stream prediction under changing distributions [8]. Recent studies further show that drift locality and magnitude affect detection difficulty and recovery time [9], that drift localization and explanation are necessary for actionable monitoring [10], and that continuous adaptation must account for predictive error, memory consumption, and runtime cost [11]. Therefore, a reliable intelligent information system should jointly track predictive performance, distribution shift, model complexity, and the estimated upper bound on future risk.

2.3. PAC-Bayes Theory and Generalization Bound

PAC-Bayes theory provides an important tool for generalization analyses in machine learning. This theory considers the prior and posterior distributions in the hypothesis space and uses the KL divergence between them to characterize the complexity changes caused by the learning process. Let P be the prior distribution before training, Q be the posterior distribution after training, $\hat{R}(Q)$ be the empirical risk, and $R(Q)$ be the true risk. Then, the common PAC-Bayes bound can be written as

$$R(Q) \le \hat{R}(Q) + \sqrt{\frac{KL(Q\|P)+\ln\frac{2\sqrt{n}}{\delta}}{2n}} \quad (1)$$

where $KL(Q\|P)$ represents the KL divergence of the posterior distribution relative to the prior distribution, n represents the sample size, and δ represents the confidence parameter. This formula shows that the upper bound of the true risk is not only the empirical risk but also the complexity of the model. When the model's empirical error is low, but the posterior distribution differs significantly from the prior, the generalization bound may be large.

PAC-Bayes theory is interpretable because it relates empirical performance to a complexity penalty between prior and posterior distributions. Viallard et al. developed practically optimizable disintegrated PAC-Bayesian bounds for individual hypotheses and demonstrated them on neural networks [1]. Sun et al. combined algorithmic

stability with PAC-Bayes analysis for multi-view learning [4]. Flynn et al. extended PAC-Bayesian analysis to lifelong learning with multi-armed bandit tasks [5].

In practice, it is not straightforward to directly apply the PAC-Bayes theory to intelligent information systems. On the one hand, large-scale neural models have large parameter dimensions, which results in a high cost estimation of the posterior distribution and the KL divergence. However, the classic PAC-Bes bound typically assumes that the training and test data are from the same distribution, which may be affected by knowledge drift in intelligent information systems. Thus, we introduce drift-aware calibration and incremental KL updates on top of the PAC-A border to adapt to dynamic knowledge environments.

2.4. Data Drift, Concept Drift, and Knowledge Update

Data drift refers to changes in the distribution of input data over time, whereas concept drift refers to changes in the relationship between input features and target labels. Drift manifests in various forms in intelligent information systems. Topic drift is characterized by changes in the distribution of knowledge categories, such as a sudden increase in the number of documents in a specific field. Lexical drift is characterized by the emergence of new terms, abbreviations, and new expressions. Entity drift is characterized by changes in the relationships between organizations, projects, products, or people. Behavioural drift is characterized by changes in user search, browsing, and clicking patterns. Tag drift is characterized by changes in the classification or recommendation criteria.

Traditional drift monitoring is no longer limited to detecting statistical differences between historical and current samples. Dong et al. showed that drift can affect the entire machine-learning pipeline rather than only the final prediction stage [6]. Bayram et al. linked concept drift to model degradation and organized performance-aware detection methods [7]. Karimian and Beigy formulated drift handling as a domain-adaptation problem and derived a generalization bound for stream prediction [8]. Aguiar and Cano demonstrated that drift locality and magnitude affect detection difficulty, classifier performance, and recovery time [9]. Hinder et al. emphasized drift localization and explanation for actionable monitoring [10], and a systematic review synthesized explainability and interpretability approaches for concept and data drift [20], whereas Chen and Dai studied continuous adaptation under memory and runtime constraints [11]. Together with broader studies of recurring and active/passive concept drift [12,13], these findings suggest that drift detection should be connected to model performance, adaptation cost, and generalization risk.

These studies form an important basis for drift detection, but they have flaws. Drift detection can detect distribution changes, but it cannot investigate whether any of the changes affect generalization. Some distribution changes are clear to observe but can be small, have irrelevant feature sizes, and have little impact on performance. Other changes may be small (in a statistical sense), and a large-model risk is likely to be generated. Drift finding must be combined with generalization risk analysis. Here, we suggest utilizing SGAM-PB, combining PAC-Bayes bound and a drift perception mechanism, to understand the influence of knowledge updates on the model's true risk.

3. SGAM-PB Model Integrating PAC-Bayes Bounds

3.1. Problem Definition

Let the dataset received by the intelligent information system at time t be:

$$D_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}. \tag{2}$$

where x_i^t represents the feature representation of the i knowledge object, user interaction, or system log, and y_i^t represents its corresponding label, feedback, relevance score, or risk category. x_i^t can be composed of text semantic vectors, knowledge entity features, metadata, user behavior sequences, temporal features, and knowledge graph structural features.

The learning algorithm is trained on the dataset D_t to obtain the prediction function f_t. The training risk is defined as

$$R_{train}(f_t) = \frac{1}{n_t}\sum_{i=1}^{n_t} \ell\,(f_t(x_i^t), y_i^t). \tag{3}$$

where $\ell(\cdot)$ is the task loss function. Deployment risk is defined as

$$R_{dep}(f_t) = \mathbb{E}_{(x,y)\sim P_{dep}^t}\,\ell(f_t(x), y). \tag{4}$$

where P_{dep}^t represents the data distribution in the real deployment environment at time t. The generalization gap is defined as

$$G_t = R_{dep}(f_t) - R_{train}(f_t). \tag{5}$$

In static learning scenarios, the training distribution is typically assumed to be consistent with the deployment distribution. However, in intelligent information systems, P_{dep}^tchanges with knowledge updates and user behavior. Therefore, this paper aims to construct a generalization analysis model capable of estimating and explaining changes in G_t in dynamic environments [18].

3.2. Overall Model Architecture

The SGAM-PB consists of seven core modules: a data acquisition layer, feature representation layer, prior-

posterior learning layer, PAC-Bayes bound estimation layer, drift-aware calibration layer, incremental KL update layer, and knowledge risk output layer. Figure 1 shows the complete skeleton of SGAM-PB. It represents a closed-loop transition between data input and risk output, in which the PAC-Bayes bound estimating module performs theoretical risk estimation, the drift calibration module performs dynamic environment adaptation, and the incremental KL updating module performs scalable computation in large scenarios.

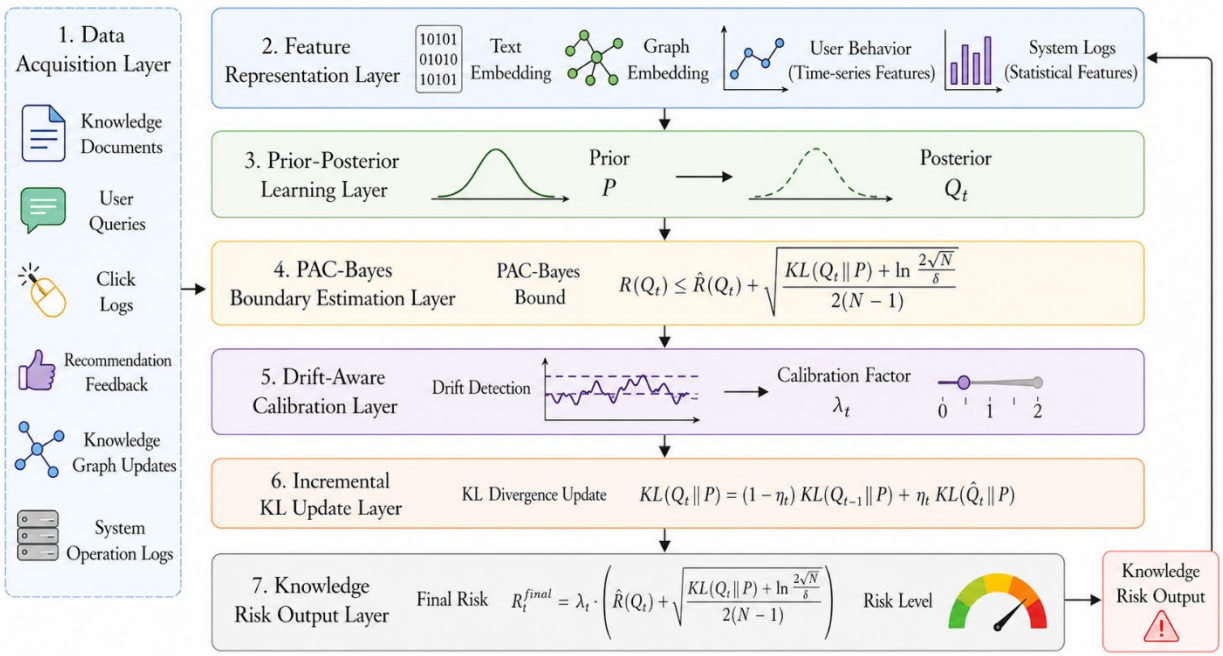

Figure 1. Overall architecture of the proposed SGAM-PB model

The data acquisition layer reads knowledge documents, user queries, click logs, recommendation reviews, knowledge graph updates, and system operation logs, and the feature representation layer translates heterogeneous data into a single vector representation. Semantic embedding models are used for context representation, knowledge objects are represented via graph embedding, user behaviour is represented using time-series features, and system logs are retrieved based on frequency, anomaly, and time-series statistics.

The prior distribution may be initialized from a historical model or learned across related tasks, while the posterior distribution is updated using the current batch. This design is consistent with PAC-Bayesian meta-learning, in which transferable prior information and task-specific posteriors are controlled through explicit generalization bounds [19]. The prior distribution P_t represents the distribution of the model's assumptions before observing the latest data, which can be constructed from historical model parameters, the knowledge state of the previous stage, or domain experience. The posterior distribution Q_t represents the distribution of the model's assumptions after learning the current data. The PAC-Bayes bound estimation layer uses empirical risk and $KL(Q_t \| P_t)$ to estimate the upper bound of the true risk [14,15].

3.3. PAC-Bayes Generalization Bound Estimation

For the posterior distribution Q_t and the prior distribution P_t, SGAM-PB defines the PAC-Bayes risk bound as follows:

$$B_{PB}(Q_t) = \hat{R}(Q_t) + \sqrt{\frac{KL(Q_t \| P_t) + \ln \frac{2\sqrt{n_t}}{\delta}}{2n_t}}, \quad (6)$$

where $\hat{R}(Q_t)$ is the empirical risk of the posterior classifier, $KL(Q_t \| P_t)$ represents the complexity of the posterior distribution relative to the prior distribution, n_t is the number of samples, and δ is the confidence parameter. This bound reflects the upper bound of the true risk of the model. If the model training error is low but the KL divergence is large, it indicates that the model may have achieved good training performance through a large hypothesis shift, and its generalization risk remains high.

In intelligent information systems, P_t can be understood as the system's original knowledge state, and Q_t can be understood as the model's state after introducing new knowledge. If Q_t changes drastically relative to P_t, it may mean that the knowledge environment has undergone significant changes, or it may mean that the model is

overfitting recent noise. Therefore, KL divergence is not only a mathematical complexity term but also an explanatory indicator of the stability of a knowledge system.

3.4. Drift-Aware Generalization Risk Score

To characterize changes in the knowledge environment, SGAM-PB defines drift intensity as

$$S_{drift}^{t} = D_{\phi}\left(P_t(X), P_{t+1}(X)\right), \quad (7)$$

where D_{ϕ} is the distribution distance metric, which can be implemented using Jensen-Shannon divergence, maximum mean difference, population stability index, or embedding distribution distance. The larger S_{drift}^{t} is, the more significant the difference between the current knowledge environment and the historical environment.

The comprehensive generalization risk score is defined as follows:

$$R_g^t = \alpha B_{PB}(Q_t) + \beta G_{emp}^t + \gamma S_{drift}^t + \lambda C_{model}^t, \quad (8)$$

where G_{emp}^t represents the empirical generalization gap, C_{model}^t represents the model complexity risk, and α, β, γ and λ are non-negative weights. This risk score unifies the theoretical generalization bound, empirical performance, drift intensity, and model complexity.

The significance of this design is that the SGAM-PB does not attempt to replace all evaluation metrics with a single indicator but rather constructs a comprehensive risk interpretation framework. When the accuracy remains stable but the drift intensity and PAC-Bayes bound increase, the system can detect potential risks in advance. When the accuracy decreases, but the drift intensity is low, the system can determine that the problem may stem from model overfitting or label noise. When the KL divergence suddenly increases, the system can determine that the model hypothesis space has undergone a significant change.

3.5. Incremental KL Update Mechanism

In large-scale intelligent information systems, recalculating the complete KL divergence after each knowledge base update incurs a high overhead. To improve scalability, the SGAM-PB introduces an incremental KL update mechanism as follows:

$$KL_{t+1} = KL_t + \Delta KL(D_t, D_{t+1}). \quad (9)$$

where $\Delta KL(D_t, D_{t+1})$ represents the complexity increment brought by the new batch of data. This mechanism treats knowledge updating as a continuous process, eliminating the need to estimate the complete difference between the posterior and prior distributions from scratch.

For neural network models, the posterior distribution can be approximated as a diagonal Gaussian distribution or a low-rank covariance structure. Let $Q_t = \mathcal{N}(\mu_t, \Sigma_t), P_t = \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$, then the KL divergence can be approximated by changes in the mean and covariance. The low-rank approximation can further reduce the storage and computational costs in high-dimensional parameter spaces.

3.6. Model Output and Decision Support

The SGAM-PB outputs five categories of results. The first category includes prediction performance metrics, such as accuracy, F1 score, AUC, and recommendation metrics. The second category is generalization behaviour metrics, including training risk, deployment risk, and generalization gaps. The third category is theoretical risk metrics, including the PAC-Bayes bound, KL divergence, and bound tightness. The fourth category is drift behaviour metrics, including topic drift, lexical drift, entity drift, and user behaviour. The fifth category is system decision metrics, including risk levels, affected modules, suggested retraining scopes, and knowledge governance measures.

For example, when the system detects high-intensity drift in a knowledge topic together with increases in the PAC-Bayes bound and KL divergence, it flags a risk of generalization failure. Because harmful drift may be localized, the system can recommend retraining the affected knowledge category rather than the entire model [6,9]. If drift is weak but KL divergence is abnormally high, SGAM-PB treats the pattern as possible over-adaptation to local noise and recommends stronger regularization or data cleaning.

4. Experimental Design

4.1. Experimental Objectives

This experiment was not conducted to compare the accuracy of the two algorithms but to test whether SGAM-PB can provide interpretable generalization ability analysis in intelligent information systems. Five questions were included: 1) Can the model provide a stable predictive performance? 2) Can it reduce the generalization gap between the training and deployment risks? 3) Can PAC-Bayes bound correctly represent the actual generalization error? 4) Can our model deliver risk responses prior to knowledge drift, or is the model scalable to large-scale data and continuous updates? 3) How should the training and deployment data be generated? 4) Which baseline models and SGAM have been trained? 5) The empirical risk, PAC-Bayes bound, KL divergence, and drift strength are calculated. 6) Can we analyse the stability of the model with knowledge drift under dynamic updates and scale expansion?

4.2. Experimental Task Scenarios

This study constructs five types of intelligent information system task scenarios, as shown in Table 1.

Table 1. Experimental scenarios and evaluation purposes

Task Scenario	Data Objects	Main purpose	Core evaluation indicators
Knowledge document classification	Corporate documents, research abstracts, patent texts	Validate knowledge classification generalization ability	Accuracy, F1, Gap
User behaviour prediction	Search, browse, click, bookmark logs	Validating user interest drift adaptability	AUC, F1, Drift Score
Knowledge Recommendation	User-knowledge item interaction, knowledge graph	Validating sparse feedback and cold start generalization	AUC, NDCG, Gap
Risk Identification	Abnormal knowledge links, error tags, expired documents	Verify the relationship between generalization risk and system error	F1, Bound, Risk Score
Dynamic knowledge updates	New themes, new entities, new terms, new tags	Verify the robustness of knowledge updates	Recovery, KL, Bound

Knowledge document classification. This system needs to classify enterprise documents, research abstracts, patent texts, policy documents, and technical reports by topic. This task was used to evaluate the generalization ability of the model for different knowledge topics and document semantics. User behaviour prediction. This user predicts his/her next knowledge needs based on his/her past search, browsing, clicking, saving, and downloading behaviour. This was used to verify that the model was stable when user interests changed. Knowledge recommendation. This test recommends documents, experts, projects, and resources based on user profiles, knowledge content, interaction history, and knowledge graph relationships. This study was used to evaluate the generalization ability of a model under sparse feedback and cold-start conditions. Risk identification. This must detect abnormal knowledge links, incorrect tags, expired documents, inconsistent entity relationships, and low-reliability recommendation results. It is used to verify that the generalization risk scores are correlated with the actual system errors. Dynamic knowledge updates. This works for every new topic, entity, word, and tag added to the knowledge base, observing the risk limits of the models, drift response, and recovery speed. This performs best in the real-world operation of intelligent systems.

4.3. Comparison Algorithms and Evaluation Metrics

We considered six types of comparison methods. Experimental risk minimization methods are basic deterministic learning baselines. Dropout regularization assesses the effect of stochastic regularization on generalization performance. Weight decay is used to evaluate the performance of classical complexity-limited methods. Deep ensemble methods are used to assess the reliability of ensemble uncertainty estimation. Bayesian neural networks were used to evaluate the probabilistic learning baseline, and the classic PAC-Bayes baseline was used to investigate the effectiveness when only PAC-Bayes bounds were used without varying drift calibration and incremental KL updates. The proposed SGAM-PB method was compared with the complete method.

The evaluation metrics were divided into five categories. The first category is prediction performance metrics, including accuracy, F1 score, and AUC. The second category is generalization metrics, including training risk, deployment risk, and generalization gaps. The third category is PAC-Bayes metrics, including the bound value, KL divergence, and bound tightness. The fourth category is drift metrics, which include drift strength, risk response latency, and update recovery speed. The fifth category is scalability metrics, including training time, bound estimation time, memory usage, and incremental-update overhead. Bound compactness is defined as

$$T_{bound} = |B_{PB} - G_{real}|. \tag{10}$$

where B_{PB} represents the PAC-Bayes bound, and G_{real} represents the generalization gap between real and observed data. The smaller the T_{bound}, the closer the bound is to the real generalization risk, and the more practically valuable the theoretical estimate is. Table 2 shows the settings of the methods compared in this study.

Table 2. Baseline algorithms and main characteristics

Algorithm	Complexity control	Drift perception	Risk Explanation	Scalability
ERM	weak	powerful	weak	high
Dropout	middle	powerful	middle	middle
Weight Decay	middle	powerful	middle	high

Algorithm	Complexity control	Drift perception	Risk Explanation	Scalability
Deep Ensemble	middle	weak	middle	Low
BNN	powerful	weak	powerful	Low
PB Baseline	powerful	weak	powerful	middle
SGAM-PB	powerful	powerful	powerful	high

5. Experimental Results and Analysis

5.1. Overall Predictive Performance Analysis

The performances of the different algorithms for the five types of smart information system tasks are shown in Figure 2. The SGAM-PB model was relatively stable. Compared to ERM, SGAM does not always achieve the highest accuracy in static tasks, but can achieve higher accuracy in cross-domain drift and knowledge updates. The EBM method performs well under the same training and test distributions but is not particularly successful once a new topic and entity are added. Dropout and weight decay can reduce overfitting to some degree, but poorly explain the generalization risk. Deep ensemble methods can improve uncertainty estimation; however, they are expensive. Although Bayesian neural networks can produce probabilistic predictions, their training time is large for large-scale information systems. The PAC-Bayes baseline can provide theoretical limits but does not explicitly model knowledge drift.

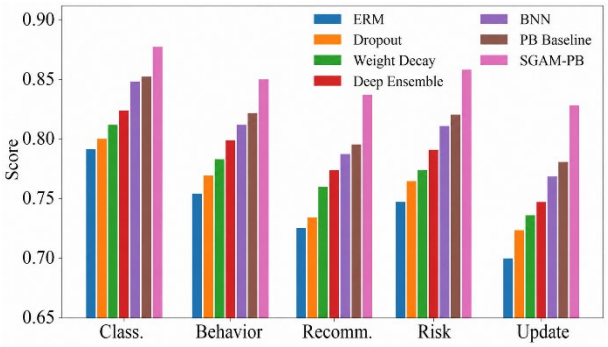

Figure 2. Overall Performance of Different Algorithms in Five Types of Intelligent Information System Tasks

Another advantage of SGAM-PB is that it is not restricted to prediction performance, generalization stability, risk interpretation, or computation cost, but rather to the highest accuracy possible. Intelligent information systems seek not only to perform well on the current set of test data, but also to be reliable after knowledge updates, user updates, and drift.

5.2. Generalization Gap Analysis

Generalization gap experiments revealed that SGAM-PB minimizes training and deployment risks in most tasks. Figure 3 shows the generalization gap between the algorithms. ERM (which largely optimizes empirical risk on training data) is prone to overfitting to historical knowledge and errors in deployment. Dropout and weight decay can decrease the risk of overfitting, but they cannot determine the model complexity and upper bound of the true risk exactly. The SGAM PBP includes both empirical risk and KL complexity to jointly optimize and evaluate via the PAC-Bayes bound, so that the generalization gap is controlled more effectively.

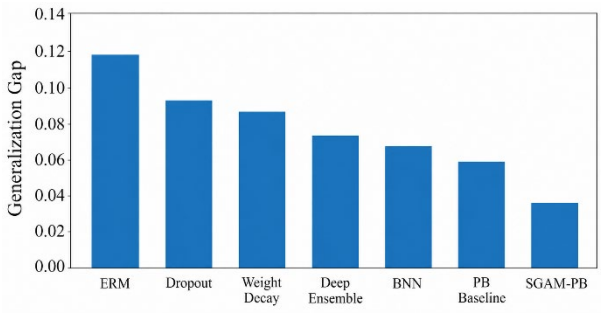

Figure 3. Comparison of Generalization Gap of Different Algorithms

The generalization gap is of practical importance in knowledge management systems. A large generalization gap in a knowledge classification model implies that the model is only valid for old knowledge; a large generalization gap in a learning recommendation model implies that the model might overlook old click patterns; a large generalized gap in a risk identification model implies that the system cannot recover new abnormal knowledge relationships. Consequently, the generalization gap can be used as a monitoring indicator before and after the adoption of smart information system models.

5.3. PAC-Bayes Bound Convergence Analysis

Figure 4 shows the convergence of the PAC-Bayes bound during training. We show that the SGAM-PB can achieve a smoother and more stable risk bound during training. In early training, the empirical risk decreases rapidly, and the bound is smaller. In the middle, the actual risk decreases rapidly, but the KL divergence increases as the model

complexity increases. In late training, the SGPM-PB gradually converges to the bound through complexity constraints and drift calibration.

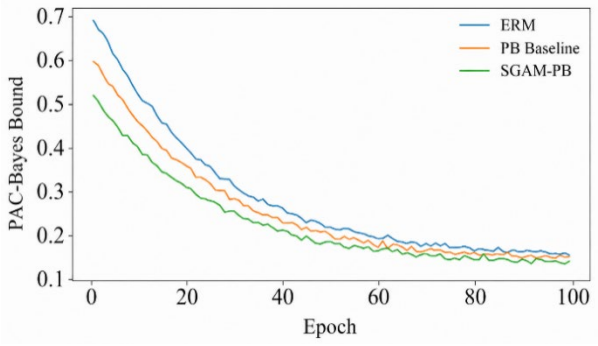

Figure 4. Convergence Curve of PAC-Bayes Bound as a Permutation of Training Rounds

This finding confirms that the PAC-Bayes bound can show phenomena that traditional accuracy measures cannot. Some models may continue to improve their accuracy even as they become higher and higher. The lower bound of the true risk remains unclear; thus, the model might fit the training sample to more complicated assumptions rather than obtaining a more stable generalization score. SGAM-PB can identify this risk and represent it in the overall generalization risk score.

5.4. Bound Validity and Tightness Analysis

Figure 5 presents the relationship between the true generalization gap and the PAC-Bayes bound. It can be observed that the bound estimated by SGAM-PB is more significant than the true standard generalization error. While the traditional PAC-Bayes baseline can provide some upper bound for prediction, the bound may be too loose or lag the correct response in knowledge drift settings. Drift-aware calibration makes the estimate closer to the true risk in dynamic environments.

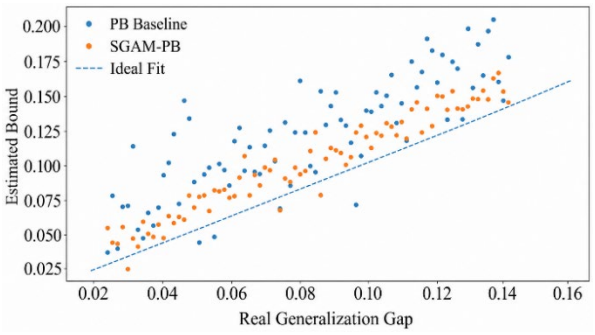

Figure 5. Relationship between the true generalization gap and the PAC-Bayes bound

Figure 6 shows a box plot of the bound tightness. We show that the T-bound distribution of the SGAM-PB is more concentrated with fewer outliers, indicating that its bound estimation is more reliable. Bound tightness is important for practical applications. If it is too loose, it cannot lead to effective decision-making by administrators; however, if it is closer to the true generalization error, it may be a reliable indicator of model deployment risk.

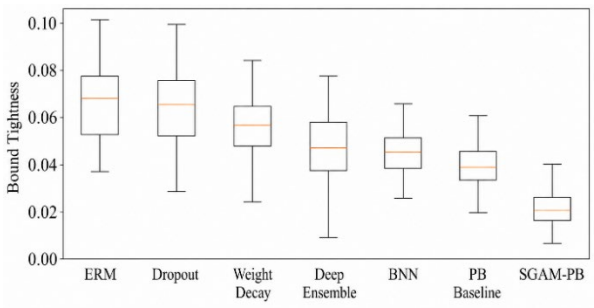

Figure 6. Relationship between Real Generalization Gap and PAC-Bayes Bound

5.5. KL Divergence and Posterior Complexity Analysis

Figure 7 shows the KL divergence and posterior complexity during training. The larger the capacity of the models, the larger the posterior shift of the posterior distribution from the prior. Large model sizes often produce large KL divergences in small samples, noisy labels, or strong drift situations. These models can perform well on the training set, although they are quite loose in PAC-Bayes, implying a larger real risk.

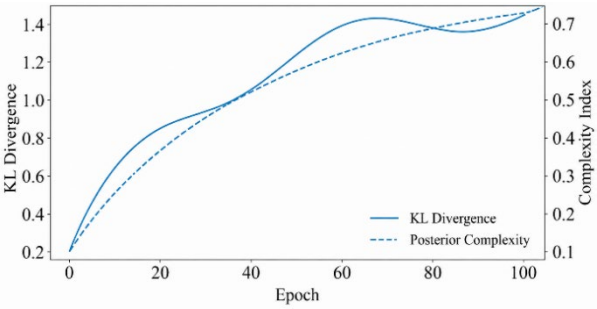

Figure 7. Evolution of KL Divergence and Posterior Complexity during Training

The SGAM-PB views KL divergence as a measure of model complexity and information state. A moderate increase in KL divergence when the knowledge base is well updated can be interpreted as showing that the model is absorbing knowledge. A sudden, sharp increase in the KL

divergence without a drastic improvement in prediction performance may indicate that the model was either too old or had too high data to adapt properly. KL divergence can help the system manager if the model updates are accurate.

5.6. Knowledge Drift Robustness Analysis

The knowledge drift experiment set different drift intensities, such as small topic changes, small entity updates, strong semantic flows, and significant user changes. Fig. 8 shows the accuracy decay of the different algorithms as drift intensification increases. We show that the performance of all algorithms decreases as the drift intensity increases, whereas SGAM-PB drops the slowest. The ERM is most sensitive to drift because it does not have an explicit complexity control and drift response. Dropout and weight decay can mitigate some overfitting, but we cannot know the source of drift. Deep ensembles and Bayesian neural networks may produce uncertainty, but cannot differentiate noise uncertainty from knowledge drift uncertainty.

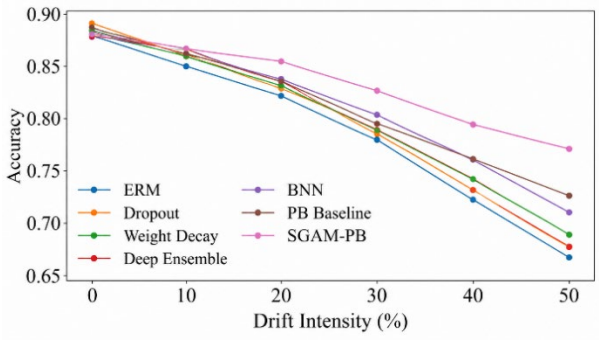

Figure 8. Accuracy decay of different algorithms with increasing drift intensity

Fig. 9 also shows the risk response under different drift values. The advantage of SGAM-PB is that the larger the drift value, the larger the overall risk score. Even if the accuracy does not drop significantly, the drift index and PAC-Bayes bound could increase ahead, and it could provide an early warning for a smart information system, as real systems would never wait until users receive incorrect advice or misclassifications.

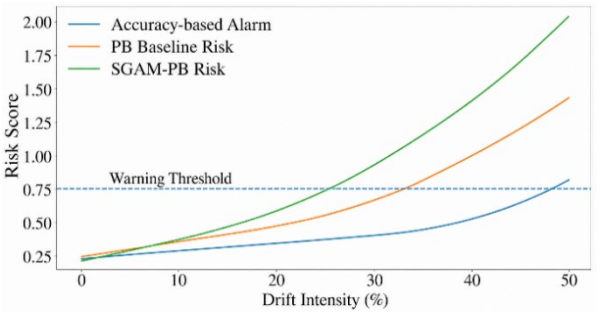

Figure 9. Risk Response under Different Drift Intensities

5.7. Incremental Knowledge Update Recovery Analysis

In the dynamic knowledge update experiment, new topics, entities, and labels were added, and the recovery rates of the models were tracked. Figure 10 shows the recovery curves for the different methods after adding new knowledge. The results show that the SGAM-PB can recover to a stable state after a knowledge update because the incremental KL update does not completely re-estimate the complexity, and the drift calibrator knows the affected knowledge region.

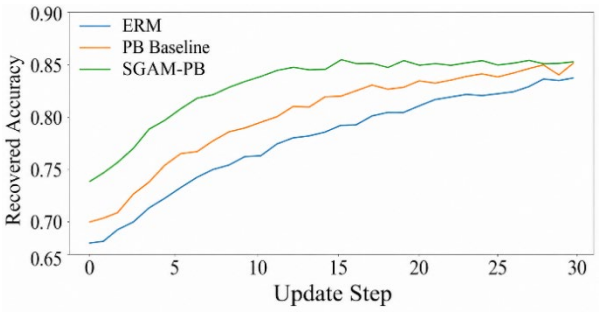

Figure 10. Recovery curves of different methods after the addition of new knowledge

This means that SGAM-PB can be used off-line and monitored with increasing accuracy. For large-scale knowledge systems, full retraining with each update may be expensive. SGAM can help decide if full retraining, local fine-tuning, or risk labelling is needed; if the drift lies in the local knowledge category, it can be considered to update the relevant modules, and if the drift stretches to multiple knowledge domains, then it could be recommended to complete retraining.

5.8. Scalability Analysis

Figure 11 shows the operating costs of all methods for higher data scales. When the data scale grows from small samples to large simulated information system data, the SGAM-PB is lower than simple ERM methods, but lower than deep ensemble and partially Bayesian neural networks. Because of the incremental KL updates and low-rank posterior approximation, the SGAP-PB bound estimation time increases smoothly with a larger data scale, and the memory usage remains within acceptable ranges.

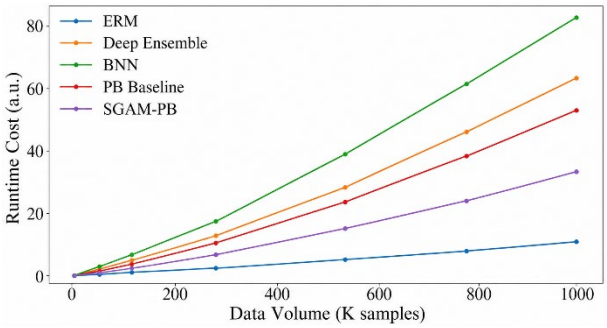

Figure 11. Operating Costs of Different Methods Under Increased Data Scale

Scalability is crucial for intelligent information systems. Knowledge management systems often contain many documents, users, entities, and records. If the cost of generalized risk analysis methods is too high, they do not have practical applications. The experimental results show that although the SGAM-PB has good engineering usability, it offers a theoretical risk interpretation.

5.9. Ablation Experiment Analysis

Figure 12 presents the ablation experiment results for each SGAM-PB module. After removing PAC-Bayes, the model only relied on the performance and drift indicators and could not estimate the true risk. After eliminating the drift-aware calibration module, the risk response is slow in knowledge updates. After replacing the KL update module, their computational cost is quite large and difficult to adapt to large-scale continuous updates. Once the risk fusion module is removed, the indicators are still unclear, and system administrators cannot make concrete decisions.

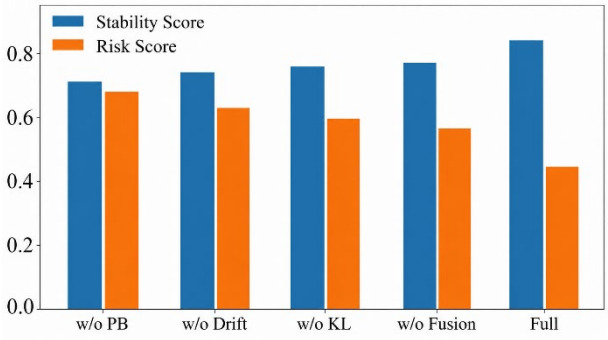

Figure 12. Ablation experiment results for each module of SGAM-PB

The complete SGAM-PB achieved the best overall performance in terms of prediction performance, generalization stability, risk interpretation, drift response, and computational efficiency. This indicates that the modules proposed in this study are not simply superimposed but rather form a complementary relationship between theoretical estimation, dynamic calibration, and engineering deployment.

Table 3 summarizes the experimental trends of SGAM-PB compared with the main comparative methods.

Table 3. Summary of experimental comparison

Method	Generalization gap	Bound compactness	Drift response	Update recovery	Computational overhead
ERM	high	none	slow	slow	Low
Dropout	middle	none	middle	middle	middle
Weight Decay	middle	none	middle	middle	Low
Deep Ensemble	middle	middle	middle	middle	high
BNN	middle	middle	middle	middle	high
PB Baseline	lower	middle	middle	middle	middle
SGAM-PB	lowest	Most Tight	Fastest	Fastest	Acceptable

6. Conclusions

This study proposes a scalable intelligent information system machine learning generalization capability analysis model, SGAM-PB, which includes PAC-Bayes bounds. Given continuous knowledge updates, user behaviours, semantic distribution drift, and difficulty understanding model deployment risks in intelligent information systems, this model unifies empirical risk, PAC-Bayes generalization limits, prior-posterior complexity, knowledge drift level, and incremental KL update as an interpretable risk analysis model. Simulations have shown that the SGAM can yield stable prediction performance for multiple intelligent information application tasks, effectively bridge the generalization gap, achieve a tighter PAC-Bayes risk bound, generate risk responses early when knowledge drift occurs, and perform at an acceptable computational time for large-scale data. Compared with existing empirical risk assessment methods, the SGPM-PB is more suitable for long-running knowledge management systems and dynamic intelligent information platforms.

The research could be conducted in three directions: (1) tighter PAC-Bayes bounds for large language models, graph neural networks, and multimodal knowledge systems; (2) causal drift detection was compared with SGAM-PB to distinguish harmful drift from harmless distribution changes; and (3) long-term deployments and evaluations were performed for real enterprise knowledge management systems, digital libraries, and intelligent recommendation frameworks to further validate the model's performance in real environments.

References

[1] Viallard, P., Germain, P., Habrard, A., & Morvant, E. A general framework for the practical disintegration of PAC-Bayesian bounds. Machine Learning. 2024; 113(2): 519–604. https://doi.org/10.1007/s10994-023-06391-0.

[2] Biggs, F., & Guedj, B. Differentiable PAC-Bayes objectives with partially aggregated neural networks. Entropy. 2021; 23(10): 1280. https://doi.org/10.3390/e23101280.

[3] Mai, T. T. Misclassification bounds for PAC-Bayesian sparse deep learning. Machine Learning. 2025; 114:18. https://doi.org/10.1007/s10994-024-06690-0.

[4] Sun, S., Yu, M., Shawe-Taylor, J., & Mao, L. Stability-based PAC-Bayes analysis for multi-view learning algorithms. Information Fusion. 2022; 86–87, 76–92. https://doi.org/10.1016/j.inffus.2022.06.006.

[5] Flynn, H., Reeb, D., Kandemir, M., & Peters, J. PAC-Bayesian lifelong learning for multi-armed bandits. Data Mining and Knowledge Discovery. 2022; 36(2): 841–876. https://doi.org/10.1007/s10618-022-00825-4.

[6] Dong, S., Wang, Q., Sahri, S., Palpanas, T., & Srivastava, D. Efficiently mitigating the impact of data drift on machine learning pipelines. Proceedings of the VLDB Endowment. 2024; 17(11): 3072–3081. https://doi.org/10.14778/3681954.3681984.

[7] Bayram, F., Ahmed, B. S., & Kassler, A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*. 2022; 245: 108632. doi: 10.1016/j.knosys.2022.108632.

[8] Karimian, M., & Beigy, H. Concept drift handling: A domain adaptation perspective. *Expert Systems with Applications*. 2023; 224: 119946. doi: 10.1016/j.eswa.2023.119946.

[9] Aguiar, G. J., & Cano, A. A comprehensive analysis of concept drift locality in data streams. *Knowledge-Based Systems*. 2024; 289: 111535. doi: 10.1016/j.knosys.2024.111535.

[10] Hinder, F., Vaquet, V., & Hammer, B. One or two things we know about concept drift—a survey on monitoring in evolving environments. Part B: Locating and explaining concept drift. *Frontiers in Artificial Intelligence*. 2024; 7: 1330258. doi: 10.3389/frai.2024.1330258.

[11] Chen, Y., & Dai, H.-L. Concept drift adaptation with continuous kernel learning. *Information Sciences*. 2024; 670: 120649. doi: 10.1016/j.ins.2024.120649.

[12] Suárez-Cetrulo, A. L., Quintana, D., & Cervantes, A. A survey on machine learning for recurring concept drifting data streams. Expert Systems with Applications. 2023; 213, 118934. https://doi.org/10.1016/j.eswa.2022.118934.

[13] Han, M., Chen, Z., Li, M., Wu, H., & Zhang, X. A survey of active and passive concept drift handling methods. Computational Intelligence. 2022; 38(4): 1492–1535. https://doi.org/10.1111/coin.12520.

[14] Chugg, B., Wang, H., & Ramdas, A. A Unified Recipe for Deriving (Time-Uniform) PAC-Bayes Bounds. *Journal of Machine Learning Research*. 2023; 24(372): 1–61. http://jmlr.org/papers/v24/23-0401.html.

[15] Rodríguez-Gálvez, B., Thobaben, R., & Skoglund, M. More PAC-Bayes bounds: From bounded losses, to losses with general tail behaviors, to anytime validity. *Journal of Machine Learning Research*. 2024; 25(110): 1–43. http://jmlr.org/papers/v25/23-1360.html.

[16] Dupuis, B., Viallard, P., Deligiannidis, G., & Simsekli, U. Uniform Generalization Bounds on Data-Dependent Hypothesis Sets via PAC-Bayesian Theory on Random Sets. *Journal of Machine Learning Research*. 2024; 25(409): 1–55. http://jmlr.org/papers/v25/24-0605.html.

[17] Tang, H., & Liu, Y. Information-Theoretic Generalization Bounds for Transductive Learning and its Applications. *Journal of Machine Learning Research*. 2024; 25(407): 1–69. http://jmlr.org/papers/v25/23-1368.html.

[18] Hou, S., Kassraie, P., Kratsios, A., Krause, A., & Rothfuss, J. Instance-Dependent Generalization Bounds via Optimal Transport. *Journal of Machine Learning Research*. 2023; 24(349): 1–51. http://jmlr.org/papers/v24/22-1293.html.

[19] Rothfuss, J., Josifoski, M., Fortuin, V., & Krause, A. Scalable PAC-Bayesian Meta-Learning via the PAC-Optimal Hyper-Posterior: From Theory to Practice. *Journal of Machine Learning Research*. 2023; 24(386): 1–62. http://jmlr.org/papers/v24/22-1254.html.

[20] Pelosi, D., Cacciagrano, D., & Piangerelli, M. Explainability and interpretability in concept and data drift: A systematic literature review. Algorithms. 2025; 18(7): 443. https://doi.org/10.3390/a18070443.