SCAMFS: STATE-LEVEL CAUSAL ASYMMETRY MODELING FOR MULTI-LABEL FEATURE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-label feature selection aims to identify compact and informative feature subsets from data with multiple correlated labels. Existing methods typically evaluate feature relevance at the label-variable level, treating each binary label as an indivisible unit. This may obscure state-specific patterns, especially when rare positive states are overwhelmed by dominant negative states, resulting in state-level dilution and label blocking. To address this issue, we propose **SCAMFS**, State-level Causal Asymmetry Modeling for Multi-label Feature Selection, as a causally motivated state-level feature selection framework. SCAMFS introduces a state-level decomposition strategy that decomposes each label into state-specific events, and further defines State-Specific Mutual Information (SSMI) and Differential State Mutual Information (DSMI) to model feature–state relevance and cross-label state dependencies. It also employs an explanatory competition mechanism to recover informative features masked by correlated label states. Experiments on seven real-world datasets show that SCAMFS outperforms representative baselines on most metrics. The anonymous code repository is available at: `https://anonymous.4open.science/r/SCAMFS-code-0C33`.

Keywords: Feature selection, multi-label, asymmetry modeling, state-level decomposition, explanatory competition.

1 INTRODUCTION

Feature selection has emerged as a fundamental technique to combat the curse of dimensionality by identifying a compact subset of relevant features (Ling et al., 2026; Li et al., 2026; Kamalov et al., 2023; Li et al., 2023a; Guyon & Elisseeff, 2003). By removing redundant and irrelevant attributes, this process significantly enhances model interpretability, reduces computational overhead, and improves generalization performance (Ling et al., 2019; Guo et al., 2022; Ling et al., 2024). Multi-label feature selection (MLFS) extends these principles to more complex data structures where each instance is simultaneously associated with multiple labels, a paradigm that has become pervasive in domains such as image annotation and bioinformatics Zhang & Zhou (2013); Gibaja & Ventura (2015); Wang et al. (2020); Yu et al. (2021); Gao et al. (2021). Effective MLFS must not only identify features relevant to individual targets but also account for the intricate dependencies and correlations within the label space Zhang & Zhou (2007); Xu et al. (2016); Wu et al. (2020). To this end, recent studies have incorporated causal-structure learning ideas Chu et al. (2023); Ling et al. (2022) for optimal spouse identification and leveraged neural feature attribution for label-specific selection, maintaining robust predictive power in high-dimensional settings. To achieve greater environmental robustness and structural stability, multi-label feature selection methods inspired by local dependency modeling have received increasing attention Huang et al. (2018b); Zhang et al. (2019); Hu et al. (2022); Zhang et al. (2023); Li et al. (2023b); Faraji et al. (2024); Dai et al. (2024); Ma et al. (2025a;b). In causal graphical settings, this line of work often relates feature selection to Markov blanket-style local dependency structures Tsamardinos et al. (2003); Guyon et al. (2007); Jiao et al. (2023), which aim to identify compact feature subsets that render the labels conditionally independent of other variables under appropriate assumptions Yu et al. (2021). By identifying such local dependency structures, MLCFS aims to provide a more stable feature subset compared to traditional correlation-based methods.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

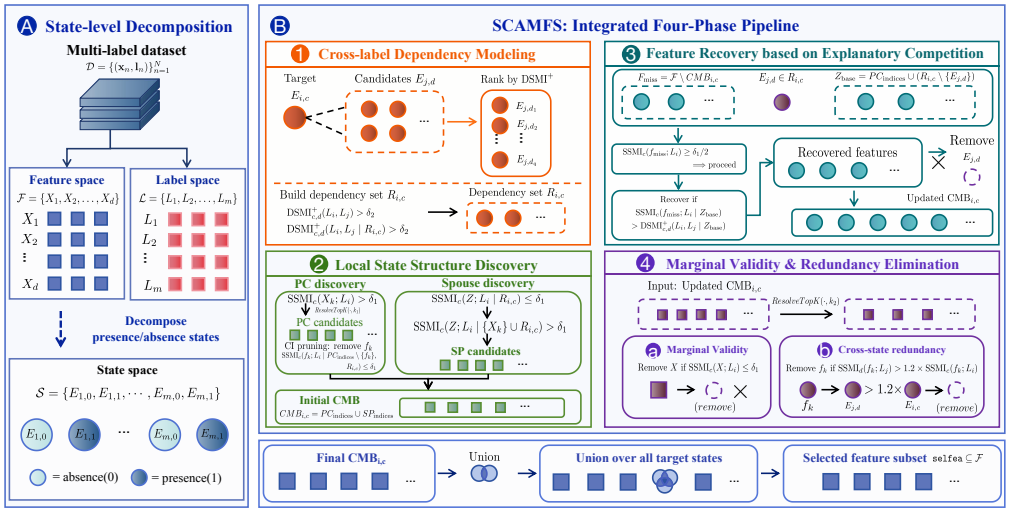


Figure 1: Overview of SCAMFS. (A) State-level decomposition: the framework first decomposes each binary label into two state-level nodes (presence and absence), forming the state space. (B) Integrated four-phase pipeline: (1) *cross-label dependency modeling* ranks candidate states by $DSMI$ and constructs an initial skeleton of state-level dependencies; (2) *local state structure discovery* identifies parent-child (PC) and spouse (SP) features via $SSMI$-based tests to build the initial CMB; (3) *feature recovery by explanatory competition* recovers previously suppressed features by competing missed features with blocking states and updates the CMB; (4) *marginal validity and redundancy elimination* refine the CMB by enforcing marginal validity and removing cross-dimensional redundancy. Finally, SCAMFS unions the refined state-level Markov blankets over all target states to obtain the final selected feature subset.

Despite the aforementioned advances, most existing multi-label feature selection methods still rely on variable-level label modeling. They treat each binary label as an indivisible unit and evaluate feature relevance with respect to the whole label variable. The motivation figure in Appendix A illustrates this limitation. As shown in Fig. 4(a), existing methods compress the presence and absence states of a label into a single variable-level representation. Consequently, heterogeneous state-specific patterns are averaged out, which may cause state-level dilution. This issue is particularly evident in multi-label data, where positive label states are often sparse and absent states dominate the empirical distribution. Traditional variable-level metrics, such as mutual information $I(X;L)$, therefore behave as weighted averages over different label states. As a result, a feature that is highly informative for a rare present state, such as X_3 in Fig. 4(a), may receive only a weak global relevance score and be mistakenly removed. Moreover, multi-label data contain complex cross-label dependencies that may also be state-specific and asymmetric. For example, the presence of one label may be strongly associated with the absence, rather than the presence, of another label. Such fine-grained interactions are difficult to capture with variable-level dependency modeling and may induce a label blocking effect, where a correlated label state absorbs or masks the contribution of a weak but informative feature. In contrast, Fig. 4(b) shows the key intuition of state-level modeling: decomposing labels into presence and absence events allows feature–state relevance and asymmetry state–state dependencies to be modeled separately.

Building on this state-level perspective, we propose **S**tate-level **C**ausal **A**symmetry **M**odeling for Multi-label **F**eature **S**election (**SCAMFS**). The key idea is to move beyond variable-level label evaluation and model state-specific explanatory relevance at the state level. As illustrated in Fig. 4(b), SCAMFS decomposes each binary label into two state-specific events, corresponding to the presence and absence states. It then uses SSMI and DSMI to measure feature–state relevance and cross-label state dependency, respectively. This design enables SCAMFS to distinguish heterogeneous state-specific patterns, mitigate state-level dilution and label blocking, and recover informative features that may be obscured by variable-level averaging. The main contributions of this paper are summarized as follows:

- **State-level asymmetry modeling for multi-label feature selection.** We propose SCAMFS, which decomposes binary labels into state-specific events and models feature relevance at the state level, providing a finer-grained alternative to conventional variable-level label modeling.

- **State-sensitive information metrics.** We introduce State-Specific Mutual Information (SSMI) and Differential State Mutual Information (DSMI) to quantify feature–state relevance and cross-label state dependency, allowing SCAMFS to capture asymmetric interactions between label states.

- **Explanatory competition and structural refinement.** We design an explanatory competition mechanism to recover features whose signals are masked by correlated label states. We further incorporate marginal validity checking and cross-state redundancy elimination to obtain a compact and discriminative feature subset.

- **Empirical effectiveness and practical efficiency.** Experiments on seven real-world multi-label datasets show that SCAMFS achieves strong performance across multiple evaluation metrics while maintaining practical computational cost.

2 Related Work

Existing MLFS methods can be broadly categorized into four groups based on their underlying mechanisms: information-theoretic, regularization-based, manifold learning-based, and Bayesian network (BN) structure learning-based methods Pereira et al. (2018); Gonzalez-Lopez et al. (2020); Wu et al. (2020); Ma et al. (2025b).

Information-Theoretic Methods. Lee & Kim (2015a) proposes the D2F method, which utilizes interaction information to measure dependencies among multiple variables; Lee & Kim (2015b) develops the FIMF method, which improves time efficiency through an information-theoretic feature ranking mechanism; Gonzalez-Lopez et al. (2020) proposes Euclidean Norm Maximization to select the features with the largest L_2-norm; Covert et al. (2023) develops an amortized optimization approach to greedily maximize conditional mutual information for dynamic feature selection; Ma et al. (2025b) presents MI-MCF, which prioritizes the distinct contributions of labels and features by utilizing mutual information to optimize the search process.

Regularization-based Methods. Ma et al. (2012) proposes SFUS for automatic image annotation; Faraji et al. (2024) develops the MLFS-GLOCAL method, which utilizes global relevance and local discriminative analysis to capture complex label relationships; Huang et al. (2018a) proposes the JFSC model, which integrates feature selection and classification into a unified regularization method. It selects task-relevant features and mines the shared feature subspace via sparse models, boosting annotation efficiency and accuracy through sparse constraints. Li et al. (2024) presents the ESRFS method, which leverages an elastic net-based method for highly sparse and low-redundancy feature selection.

Manifold Learning-based Methods. Cai & Zhu (2018) proposes MSSL, which incorporates feature manifold learning and sparse regularization to preserve the geometric structure of the data; Huang et al. (2018b) utilizes MCLS via Laplacian scores to handle complex multi-label data; Zhang et al. (2019) develops the MDFS method, which applies manifold regularization to construct low-dimensional embeddings for discriminative selection.

Causal and BN Structure Learning-based Methods. Beyond correlation-based methods, causal discovery methods focus on identifying the MB to ensure structural stability and interpretability. Wu et al. (2020) proposes MB-MCF, which identifies the MB for multiple labels by learning local causal structures in a unified method. Yu et al. (2021) develops M2LC, which formulates feature selection as a local causal discovery task and employs error-correction routines to handle dependencies within the label space. Furthermore, Wu et al. (2023) proposes KMB, a robust method that discovers Markov boundaries without relying on the strong faithfulness assumption, effectively restoring causal signals masked by dominant associations.

3 THE PROPOSED METHOD

This section elaborates on the SCAMFS method in detail. Fig. 1 illustrates the overall framework of SCAMFS. Designed specifically for multi-label data, SCAMFS is a causally motivated state-level asymmetric feature selection method. Its core innovation lies in advancing the granularity of dependency modeling from the traditional variable level to the fine-grained state level via state-level decomposition. For feature selection, SCAMFS performs local structure modeling using state-level conditional information scores to construct candidate local dependency structures. By defining State-Specific Mutual Information (SSMI) and Differential State Mutual Information (DSMI), SCAMFS can identify and mitigate state-level dilution and label blocking caused by heterogeneous state-specific patterns and cross-label state interactions. As outlined in the overall pseudocode in Appendix B.1 (Algorithm 1), the method consists of four collaborative phases: state-level cross-label dependency modeling, local state structure discovery, state-level feature recovery based on explanatory competition, and state-level marginal validity and redundancy elimination.

3.1 CAUSALLY MOTIVATED STATE-LEVEL ASYMMETRY AND INFORMATION METRICS

Let the multi-label dataset be $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{l}_n)\}_{n=1}^{N}$, where $\mathbf{x}_n \in \mathbb{R}^d$ denotes the feature vector and $\mathbf{l}_n \in \{0,1\}^m$ denotes the label vector. The feature space is $\mathcal{F} = \{X_1, \ldots, X_d\}$ and the label space is $\mathcal{L} = \{L_1, \ldots, L_m\}$.

Conventional multi-label feature selection usually evaluates the relevance between a feature X and an entire label variable L_j. This variable-level view implicitly treats the two states of a binary label as homogeneous manifestations of the same mechanism. However, in many multi-label systems, the presence and absence of a label may be associated with different feature patterns and different cross-label interactions. Therefore, evaluating feature relevance only at the variable level may dilute state-specific signals, especially when one label state is rare. For a binary label L_j, the variable-level mutual information admits the following decomposition, whose derivation is provided in Appendix C.1:

$$I(X; L_j) = \sum_{c \in \{0,1\}} P(L_j = c) D_{\mathrm{KL}}\big(P(X \mid L_j = c) \,\|\, P(X)\big). \tag{1}$$

Thus, a feature that is highly informative for a rare state may still contribute weakly to the global score $I(X; L_j)$ because its contribution is weighted by $P(L_j = c)$. We refer to this effect as state-level dilution.

Modeling Hypothesis 1 (State-level Asymmetry). Different states of a label may exhibit heterogeneous statistical patterns and may participate in different cross-label interactions. Therefore, feature–label relevance and label–label dependency should be modeled at the state level rather than only at the variable level.

Definition 1 (State-level Decomposition)**.** *For each label $L_j \in \mathcal{L}$ and each state $c \in \{0,1\}$, we define a state event*

$$E_{j,c} = \mathbf{1}[L_j = c]. \tag{2}$$

The state-level label space is then

$$\mathcal{S} = \{E_{j,c} : j = 1, \ldots, m,\ c \in \{0,1\}\}. \tag{3}$$

Here, $E_{j,1}$ denotes the present state of label L_j, while $E_{j,0}$ denotes its absent state.

Definition 2 (State-Specific Mutual Information)**.** *Given a feature X and a label state $L_j = c$, the State-Specific Mutual Information (SSMI) is defined as*

$$\mathrm{SSMI}_c(X; L_j) = D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big). \tag{4}$$

Given a conditioning set Z, its conditional form is

$$\mathrm{SSMI}_c(X; L_j \mid Z) = \sum_{z} P(z \mid L_j = c) D_{\mathrm{KL}}\big(P(X \mid Z = z, L_j = c) \| P(X \mid Z = z)\big). \tag{5}$$

The conditional form can be viewed as a context-wise extension of Eq. (4)*, where the state-specific distributional deviation is computed within each configuration of Z and then averaged under the target state. A detailed explanation is provided in Appendix C.3. In implementation, SCAMFS uses a normalized version by dividing the above score by the state surprise $-\log P(L_j = c)$.*

Table 1: HAMMING LOSS FOR EIGHT METHODS ON SEVEN DATASETS

ALGORITHM	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.2331**	**0.0439**	**0.2798**	**0.0292**	0.0555	**0.1640**	**0.1812**
MI-MCF	0.2956	0.0780	<u>0.2861</u>	0.0549	<u>0.0522</u>	0.1823	0.2034
M2LC	0.3054	0.0941	0.3183	0.0549	**0.0506**	0.1883	<u>0.1956</u>
D2F	0.3103	<u>0.0618</u>	0.3153	0.0407	0.0611	0.1910	0.2165
MDFS	0.3448	0.0833	0.3032	0.0481	0.0598	<u>0.1757</u>	0.2075
JFSC	<u>0.2660</u>	0.0618	0.3123	<u>0.0402</u>	0.0562	0.1920	0.2023
FIMF	0.3276	<u>0.0683</u>	0.3173	0.0441	0.0560	0.1873	0.2072
MCLS	0.3227	0.0806	0.3052	0.0423	0.0600	0.1880	0.2019

Definition 3 (Differential State Mutual Information)**.** *Given two label states $L_i = c$ and $L_j = d$, the Differential State Mutual Information (DSMI) measures the directional state-pair association:*

$$\mathrm{DSMI}_{c,d}(L_i, L_j) = P(L_j = d \mid L_i = c) \log \frac{P(L_j = d \mid L_i = c)}{P(L_j = d)}. \tag{6}$$

Given a conditioning set Z, its conditional form is

$$\mathrm{DSMI}_{c,d}(L_i, L_j \mid Z) = \sum_z P(z \mid L_i = c) P(L_j = d \mid L_i = c, z) \log \frac{P(L_j = d \mid L_i = c, z)}{P(L_j = d \mid z)}. \tag{7}$$

The conditional form can be viewed as a context-wise extension of Eq. (6)*, where the directional association from $L_i = c$ to $L_j = d$ is evaluated within each configuration of Z and then averaged under the source state. A detailed explanation is provided in Appendix C.4. SCAMFS retains the positive dependency component* $\mathrm{DSMI}^+ = \max(0, \mathrm{DSMI})$ *and normalizes it by* $-\log P(L_j = d)$*, which makes feature–state and state–state scores comparable during explanatory competition. The rationale for this state-surprise normalization is discussed in Appendix C.5.*

Unless otherwise specified, SSMI and DSMI^+ in the following algorithmic criteria denote their normalized versions.

3.2 STATE-LEVEL CROSS-LABEL DEPENDENCY MODELING

Phase 1 constructs the state-level cross-label dependency set $R_{i,c}$ for a target state $L_i = c$. For each other label L_j and each of its states $d \in \{0, 1\}$, SCAMFS forms the event vector $E_{j,d} = \mathbf{1}[L_j = d]$ and measures its positive state-pair dependency with the target event $E_{i,c}$ using normalized DSMI:

$$\mathrm{DSMI}^+_{c,d}(L_i, L_j) > \delta_2. \tag{8}$$

All candidates satisfying this criterion are ranked by their DSMI^+ scores. SCAMFS then greedily scans the ranked candidates and adds a candidate state $E_{j,d}$ into $R_{i,c}$ only if its dependency remains significant after conditioning on the currently selected state set:

$$\mathrm{DSMI}^+_{c,d}(L_i, L_j \mid R_{i,c}) > \delta_2. \tag{9}$$

The resulting $R_{i,c}$ captures informative cross-label state context for the target state and is used in the following phases. The detailed pseudocode of this phase is provided in Appendix B.2 as Algorithm 2.

3.3 LOCAL STATE STRUCTURE DISCOVERY

SCAMFS first screens features by the unconditional state-specific score $\mathrm{SSMI}_c(X; L_i)$. Features satisfying

$$\mathrm{SSMI}_c(X; L_i) > \delta_1 \tag{10}$$

are ranked in descending order, and the top $\mathrm{ResolveTopK}(|Cand_PC|, k_1)$ candidates are retained. Each retained feature is then tested under the state context formed by the other PC candidates and $R_{i,c}$:

$$\mathrm{SSMI}_c\big(X; L_i \mid (PC_{\text{indices}} \setminus \{X\}) \cup R_{i,c}\big) > \delta_1. \tag{11}$$

Features passing this conditional test form PC_{indices}.

SCAMFS then searches for spouse candidates among $\mathcal{F} \setminus PC_{\text{indices}}$. A feature $Z \notin PC_{\text{indices}}$ is selected as a spouse candidate if there exists $X \in PC_{\text{indices}}$ such that

$$\text{SSMI}_c(Z; L_i \mid R_{i,c}) \leq \delta_1 \quad \wedge \quad \text{SSMI}_c(Z; L_i \mid \{X\} \cup R_{i,c}) > \delta_1. \tag{12}$$

The output of Phase 2 is the current state-level candidate Markov blanket:

$$CMB_{i,c} = PC_{\text{indices}} \cup SP_{\text{indices}}. \tag{13}$$

This set is further refined in the subsequent recovery and redundancy elimination phases. The auxiliary definitions of state-level spouse candidates and state-specific candidate Markov blankets, together with the detailed pseudocode of this phase, are provided in Appendix C.6 and Appendix B.3, respectively.

Table 2: MICRO-F1 FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	MICRO-F1 (↑)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.7559**	**0.8864**	**0.6803**	**0.8300**	**0.6481**	**0.6273**	**0.6883**
MI-MCF	0.6859	0.8000	<u>0.6705</u>	0.6820	0.5019	0.5909	<u>0.6560</u>
M2LC	0.6720	0.7244	0.6318	0.7632	0.5544	0.5909	0.6380
D2F	0.6736	<u>0.8535</u>	0.6180	0.7613	0.5985	0.5248	0.6319
MDFS	0.5954	0.7586	0.6529	0.7091	0.5223	<u>0.6082</u>	0.6472
JFSC	<u>0.7379</u>	0.8392	0.6175	0.7685	<u>0.6039</u>	0.5636	0.6547
FIMF	0.6732	0.8252	0.6359	0.7286	0.5553	0.5868	0.6501
MCLS	0.6203	0.7857	0.6538	<u>0.7857</u>	0.4799	0.5772	0.6549

3.4 STATE-LEVEL FEATURE RECOVERY BASED ON EXPLANATORY COMPETITION

Phase 3 mitigates the blocking effect caused by correlated label states. A feature may be excluded in Phase 2 because its state-specific signal is partially explained by a highly correlated label state in $R_{i,c}$. To recover such features, SCAMFS revisits the missing feature set $\mathcal{F} \setminus CMB_{i,c}$ and keeps only candidates with weak but non-negligible marginal state relevance:

$$\text{SSMI}_c(X; L_i) \geq \delta_1/2. \tag{14}$$

For each remaining candidate feature X and each selected label state $E_{j,d} \in R_{i,c}$, SCAMFS constructs a base conditioning set:

$$Z_{\text{base}} = PC_{\text{indices}} \cup (R_{i,c} \setminus \{E_{j,d}\}). \tag{15}$$

It then compares the conditional explanatory strength of the feature with that of the label state:

$$\text{SSMI}_c(X; L_i \mid Z_{\text{base}}) > \text{DSMI}^{+}_{c,d}(L_i, L_j \mid Z_{\text{base}}). \tag{16}$$

If this inequality holds, the feature provides stronger conditional explanation for the target state than the competing label state. SCAMFS therefore adds X into $CMB_{i,c}$ and removes the corresponding blocking state from the active dependency set. For computational efficiency, the implementation adopts a single-success recovery strategy: for each target state, the recovery loop stops after the first successful feature–state replacement. The formal explanatory-competition criterion and the detailed pseudocode of this phase are provided in Appendix C.7 and Appendix B.4, respectively.

3.5 STATE-LEVEL MARGINAL VALIDITY AND REDUNDANCY ELIMINATION

The final phase refines the recovered candidate set $CMB_{i,c}$ by enforcing state-level marginal validity and removing cross-state redundancy. SCAMFS first ranks candidate features according to their state-specific relevance $\text{SSMI}_c(X; L_i)$ and retains the top $\text{ResolveTopK}(|CMB_{i,c}|, k_2)$ features. This step controls the size of the local candidate set and prioritizes features that are most informative for the target state.

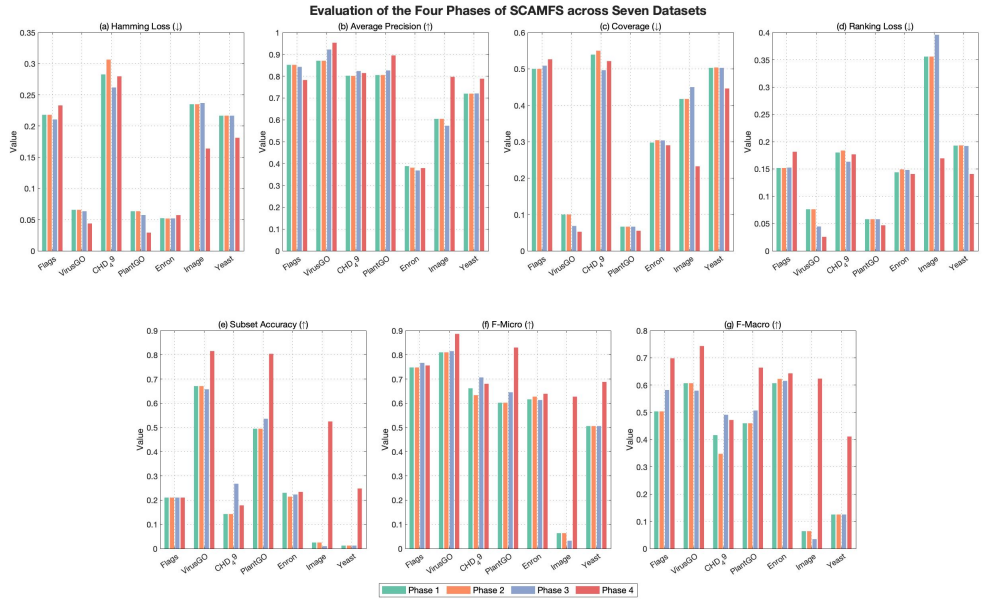


Figure 2: Ablation study of SCAMFS across four phases. The subfigures (a)–(g) illustrate the performance evolution from Phase 1 to Phase 4 across seven datasets. Note that (a), (c), and (d) are metrics where lower values indicate better performance (↓), while (b), (e), (f), and (g) are metrics where higher values indicate better performance (↑).

SCAMFS then verifies marginal validity. A feature X is retained only if

$$\mathrm{SSMI}_c(X; L_i) > \delta_1. \tag{17}$$

This removes weak state-level associations that may have been introduced during state-level recovery.

Finally, SCAMFS removes cross-state redundant features. For each retained feature X, the algorithm compares its relevance to the target state $L_i = c$ with its relevance to every non-target state $L_j = d$, where $j \neq i$. If

$$\mathrm{SSMI}_d(X; L_j) > 1.2 \times \mathrm{SSMI}_c(X; L_i), \tag{18}$$

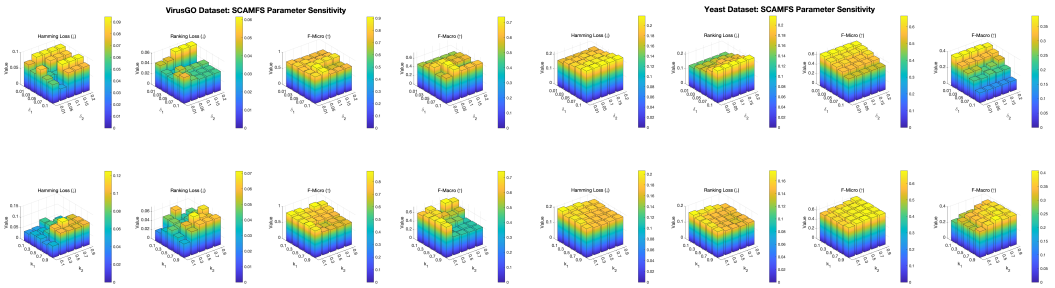
then X is considered more strongly associated with another label state and is removed from $CMB_{i,c}$. The remaining features are added to the global selected feature subset. The formal retention criterion and the detailed pseudocode of this phase are provided in Appendix C.8 and Appendix B.5, respectively.

4 Experiments

In this section, we evaluate SCAMFS on seven real-world multi-label datasets using seven metrics. We compare it with seven representative baselines from four categories: MI-based methods (MI-MCF, D2F, and FIMF), regularization-based methods (JFSC), manifold-based methods (MDFS and MCLS), and BN-structure-learning-based methods (M2LC), all introduced in Section 2. We further analyze the contribution of the four SCAMFS phases, study the sensitivity to δ_1, δ_2, k_1, and k_2 on *Yeast* and *VirusGO*, and report run-time results to assess computational efficiency.

4.1 Experimental Setup

Datasets. We evaluate SCAMFS on seven real-world multi-label datasets from diverse domains: *Flags*, *VirusGO*, *CHD_49*, *PlantGO*, *Enron*, *Image*, and *Yeast*. These datasets cover a wide range of feature dimensionalities, from 19 to 3091, and include both dense and sparse label distributions. Detailed dataset statistics, including label cardinality, label density, and the average imbalance ratio per label (*AvgIR*), are reported in Appendix E.



(a) VirusGO dataset (b) Yeast dataset

Figure 3: Parameter sensitivity analysis on VirusGO and Yeast datasets.

Evaluation Metrics. To assess the performance of the selected feature subsets, we utilize seven widely recognized metrics in multi-label learning: Hamming Loss (HL), Average Precision (AvP), Coverage (Cov), Ranking Loss (RL), Subset Accuracy (SA), F-Macro (Fma), and F-Micro (Fmi). For HL, Cov, and RL, lower values signify superior performance, while for the remaining metrics, higher values indicate better results. The formal definitions and interpretations of these metrics are provided in Appendix F.

Baselines. We compare SCAMFS against seven representative baselines: (a) MI-based: MI-MCF, D2F and FIMF; (b) Regularization-based: JFSC; (c) Manifold-based: MDFS and MCLS and (d) BN-based: M2LC.

Parameter Settings. All experiments are implemented in MATLAB R2021b and executed on a MacBook Air with an Apple M3 chip and 16 GB unified memory. For classification, we use ML-kNN Zhang & Zhou (2007) with the number of nearest neighbors fixed to $k = 10$. For SCAMFS, k_1 and k_2 are selected from $\{0.1, 0.2, \ldots, 1.0\}$ using training-only validation. The information-score thresholds δ_1 and δ_2 are selected from candidate values generated from the empirical distributions of feature–state and state–state information scores on the training data.

4.2 Comparison with Representative Baselines

This section compares SCAMFS with seven representative multi-label feature selection methods across seven datasets and seven evaluation metrics. Due to space limitations, we report two representative metrics, *Hamming Loss* and *Micro-F1*, in Tables 1–2. The complete results for the remaining five metrics, including *Subset Accuracy*, *Average Precision*, *Coverage*, *Ranking Loss*, and *Macro-F1*, are provided in Appendix G. Overall, SCAMFS achieves competitive or superior performance on most datasets and evaluation metrics.

Fig. 5 in Appendix H visualizes the normalized performance of all methods across seven datasets and seven evaluation metrics. Each subplot corresponds to one metric, and each axis denotes a dataset. All scores are mapped to $[0, 1]$, with larger values indicating better performance; for loss-based metrics, the normalized scores are reversed. The bold blue curve shows that SCAMFS generally encloses a larger area than the competing methods, especially for *Hamming Loss*, *Subset Accuracy*, *Ranking Loss*, *F-Macro*, and *F-Micro*. This indicates that SCAMFS achieves a more balanced performance profile across datasets and metrics.

To assess whether the observed performance differences are statistically significant, we conduct the Friedman test at the 95% confidence level. The null hypothesis assumes that all compared methods have equivalent average ranks across the seven datasets. As reported in Appendix I (Table 9), the Friedman statistic F_F exceeds the critical value 2.2371 for all seven evaluation metrics. Therefore, the null hypothesis is rejected for every metric, indicating statistically significant differences among the compared methods.

We further apply the Bonferroni–Dunn post-hoc test Dunn (1961) to compare SCAMFS with the baselines. The corresponding critical difference (CD) diagrams are shown in Appendix I (Fig. 6). SCAMFS consistently ranks among the top methods and achieves the best average rank on most metrics, including Hamming Loss, Subset Accuracy, Average Precision, Ranking Loss, Macro-F1,

and Micro-F1. For Coverage, SCAMFS remains competitive but does not dominate all baselines. Overall, the statistical tests confirm that SCAMFS delivers strong and reliable performance across multiple evaluation criteria.

4.3 Analysis of the Four Phases of SCAMFS

Fig. 2 shows the contribution of the four phases of SCAMFS. Phase 1 and Phase 2 produce similar results on most datasets, suggesting that the state-level cross-label context already provides a useful basis for local feature discovery, while Phase 2 mainly refines the initial candidates. The major improvement generally appears after Phase 3, especially on sparse datasets such as *VirusGO* and *PlantGO*. On *VirusGO*, Subset Accuracy rises from about 0.67 to over 0.80, and Hamming Loss drops below 0.05. This indicates that explanatory competition helps recover informative features masked by correlated label states. Phase 4 further improves the final feature subset through marginal validity checking and redundancy elimination. Its effect is particularly clear on the continuous datasets. On *Yeast*, for example, Hamming Loss decreases to around 0.18, Ranking Loss decreases to about 0.14, and Micro-F1 increases to nearly 0.69. Overall, the four phases are complementary: Phase 1–2 establish the local state structure, Phase 3 recovers blocked features, and Phase 4 yields a compact and more discriminative feature set.

4.4 Parameter Sensitivity Analysis

Fig. 3 evaluates the sensitivity of SCAMFS to δ_1, δ_2, k_1, and k_2 on *VirusGO* and *Yeast*. Overall, SCAMFS remains competitive across a broad range of settings. *VirusGO* shows higher sensitivity, especially for Hamming Loss and Ranking Loss, suggesting that sparse and imbalanced label spaces require more careful recovery and redundancy control. In contrast, *Yeast* exhibits smoother performance surfaces across most parameter combinations, particularly for Hamming Loss, Ranking Loss, and Micro-F1.

4.5 Run-Time Analysis

The complete run-time comparison is reported in Appendix J (Table 10). Overall, SCAMFS maintains a practical computational cost while performing fine-grained state-level modeling. It is particularly efficient on small datasets, achieving the fastest run time on *Flags* and *CHD_49*, and remains competitive on *VirusGO*. On larger or sparse datasets such as *PlantGO* and *Enron*, SCAMFS requires more computation because it explicitly models feature–state and state–state dependencies. Nevertheless, its run time remains practical and is much lower than the most expensive baselines, such as MI-MCF and M2LC on *Enron*.

5 Conclusion

In this paper, we proposed SCAMFS, a causally motivated state-level asymmetric feature selection method for multi-label data. By introducing SSMI and DSMI, SCAMFS moves beyond variable-level label evaluation and mitigates state-level dilution and label-blocking effects caused by heterogeneous state-specific patterns. Experiments on seven real-world datasets show that SCAMFS achieves strong and statistically reliable performance across multiple metrics with practical computational cost. Nevertheless, SCAMFS relies on empirical information estimation over discrete state events, which may be less stable under extreme label imbalance or sparse observations. Future work will explore robust state-specific information estimators, adaptive smoothing for rare states, and extensions to nonlinear representation spaces.

References

Xiaowei Cai and En Zhu. Multi-label feature selection via feature manifold learning and sparsity regularization. *Neurocomputing*, 273:487–493, 2018.

Zhixuan Chu, Ruopeng Li, Stephen Rathbun, and Sheng Li. Continual causal inference with incremental observational data. In *Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 3430–3439. IEEE, 2023.

Ian Covert, Wei Qiu, Mingyu Lu, Nayoon Kim, Nathan White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pp. 6441–6463. PMLR, 2023.

Jun Dai, Wenbin Huang, Changzhong Zhang, and Jipeng Liu. Multi-label feature selection by strongly relevant label gain and label mutual aid. *Pattern Recognition*, 145:109945, 2024. doi: 10.1016/j.patcog.2023.109945.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

Mohammad Faraji, Amjad Seyedi, Fardin Akhlaghian Tab, and Reza Mahmoodi. Multi-label feature selection with global and local label correlation. *Expert Systems with Applications*, 246:123198, 2024. doi: 10.1016/j.eswa.2024.123198.

Wanfu Gao, Yonghao Li, and Liang Hu. Multilabel feature selection with constrained latent structure shared term. *IEEE Transactions on Neural Networks and Learning Systems*, 34(3):1253–1262, 2021. doi: 10.1109/TNNLS.2021.3105142.

Eva Gibaja and Sebastian Ventura. A tutorial on multi-label learning. *ACM Computing Surveys (CSUR)*, 47(3):1–38, 2015.

Jorge Gonzalez-Lopez, Sebastián Ventura, and Alberto Cano. Distributed multi-label feature selection using individual mutual information measures. *Knowledge-Based Systems*, 188:105052, January 2020. doi: 10.1016/j.knosys.2019.105052.

Xianjie Guo, Kui Yu, Fuyuan Cao, Peipei Li, and Hao Wang. Error-aware markov blanket learning for causal feature selection. *Information Sciences*, 589:849–877, April 2022. doi: 10.1016/j.ins.2021.12.118.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Isabelle Guyon, Constantin Aliferis, and André Elisseeff. Causal feature selection. In *Computational Methods of Feature Selection*, pp. 79–102. Chapman and Hall/CRC, London, U.K., 2007.

Liang Hu, Lin Gao, Yonghao Li, Peipei Zhang, and Wanfu Gao. Feature-specific mutual information variation for multi-label feature selection. *Information Sciences*, 593:449–471, 2022. doi: 10.1016/j.ins.2022.02.016.

Jun Huang, Guozheng Li, Qiang Huang, and Xindong Wu. Joint feature selection and classification for multi-label learning. *IEEE Transactions on Cybernetics*, 48(3):876–889, 2018a.

Rong Huang, Wenbin Jiang, and Guodao Sun. Manifold-based constraint laplacian score for multi-label feature selection. *Pattern Recognition Letters*, 112:346–352, 2018b. doi: 10.1016/j.patrec.2018.08.019.

Rui Jiao, Bach Hoai Nguyen, Bing Xue, and Mengjie Zhang. A survey on evolutionary multiobjective feature selection in classification: Approaches, applications, and challenges. *IEEE Transactions on Evolutionary Computation*, 28(4):1156–1176, August 2023. doi: 10.1109/TEVC.2023.3284451.

Firuz Kamalov, Fadi Thabtah, and Ho Hon Leung. Feature selection in imbalanced data. *Annals of Data Science*, 10(6):1527–1541, 2023. doi: 10.1007/s40445-021-00366-5.

Jaesung Lee and Dae-Won Kim. Mutual information-based multi-label feature selection using interaction information. *Expert Systems with Applications*, 42(4):2013–2025, 2015a.

Jaesung Lee and Dae-Won Kim. Fast multi-label feature selection based on information-theoretic feature ranking. *Pattern Recognition*, 48(9):2761–2771, 2015b.

Peipei Li, Haixiang Zhang, Xuegang Hu, and Xindong Wu. High-dimensional multi-label data stream classification with concept drifting detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8085–8099, 2023a. doi: 10.1109/TKDE.2022.3193437.

Yan Li, Defeng Sun, and Liping Zhang. Unsupervised feature selection via nonnegative orthogonal constrained regularized minimization. *Journal of Machine Learning Research*, 27(39):1–44, 2026.

Yonghao Li, Liang Hu, and Wanfu Gao. Multi-label feature selection via robust flexible sparse regularization. *Pattern Recognition*, 134:109074, February 2023b. doi: 10.1016/j.patcog.2022.109074.

Yonghao Li, Liang Hu, and Wanfu Gao. Multi-label feature selection with high-sparse personalized and low-redundancy shared common features. *Information Processing & Management*, 61(3): 103633, 2024.

Zhaolong Ling, Kui Yu, Hao Wang, Lin Liu, Wei Ding, and Xindong Wu. Bamb: A balanced markov blanket discovery approach to feature selection. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–25, October 2019.

Zhaolong Ling, Bo Li, Yiwen Zhang, Qingren Wang, Kui Yu, and Xindong Wu. Causal feature selection with efficient spouses discovery. *IEEE Transactions on Big Data*, 9(2):1–15, 2022. doi: 10.1109/TBDATA.2022.3184614.

Zhaolong Ling, Jingxuan Wu, Yiwen Zhang, Peng Zhou, Bingbing Jiang, and Kui Yu. Causal feature selection with imbalanced data. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(2):1–13, 2024. doi: 10.1109/TETCI.2024.10633852.

Zhaolong Ling, Dongliang Liang, Xinyan Liang, Xingyu Wu, Peng Zhou, Kui Yu, and Xindong Wu. Multi-view causal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–16, 2026.

Lin Ma, Liang Hu, Qiang Huang, Pingting Hao, and Juncheng Hu. Mcf-spouse: A multi-label causal feature selection method with optimal spouses discovery. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI-25)*, 2025a.

Lin Ma, Liang Hu, Yonghao Li, Weiping Ding, and Wanfu Gao. Mi-mcf: A mutual information-based multilabel causal feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9864–9878, 2025b.

Zhigang Ma, Feiping Nie, Yi Yang, Jasper R. R. Uijlings, and Nicu Sebe. Web image annotation via subspace-sparsity collaborated feature selection. *IEEE Transactions on Multimedia*, 14(4): 1021–1030, August 2012. doi: 10.1109/TMM.2012.2191942.

Rafael B. Pereira, Alexandre Plastino, Bianca Zadrozny, and Luiz H. C. Merschmann. Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review*, 49(1):57–78, 2018. doi: 10.1007/s10462-016-9512-y.

Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander R. Statnikov. Algorithms for large scale markov blanket discovery. In *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, volume 2, pp. 376–381, 2003.

Hao Wang, Zhaolong Ling, Kui Yu, and Xindong Wu. Towards efficient and effective discovery of markov blankets for feature selection. *Information Sciences*, 509:227–242, January 2020. doi: 10.1016/j.ins.2019.09.016.

Xingyu Wu, Bingbing Jiang, Kui Yu, Huanhuan Chen, and Chunyan Miao. Multi-label causal feature selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6430–6437, 2020.

Xingyu Wu, Bingbing Jiang, Tianhao Wu, and Huanhuan Chen. Practical markov boundary learning without strong assumptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10388–10398, 2023.

Jianhua Xu, Jiali Liu, Jing Yin, and Chengyu Sun. A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. *Knowledge-Based Systems*, 98:172–184, 2016.

Kui Yu, Mingzhu Cai, Xingyu Wu, Lin Liu, and Jiuyong Li. Multilabel feature selection: A local causal structure learning approach. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3193–3207, 2021. doi: 10.1109/TNNLS.2021.3117585.

Jing Zhang, Zhongyuan Luo, Chao Li, Cheng Zhou, and Shaozi Li. Manifold regularized discriminative feature selection for multi-label learning. *Pattern Recognition*, 95:136–150, 2019. doi: 10.1016/j.patcog.2019.06.003.

Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

Panpan Zhang, Genlin Liu, and Jun Song. Mfsjmi: Multi-label feature selection considering join mutual information and interaction weight. *Pattern Recognition*, 138:109378, June 2023. doi: 10.1016/j.patcog.2023.109378.

A Motivation: State-Level Causal Asymmetry and Label Blocking

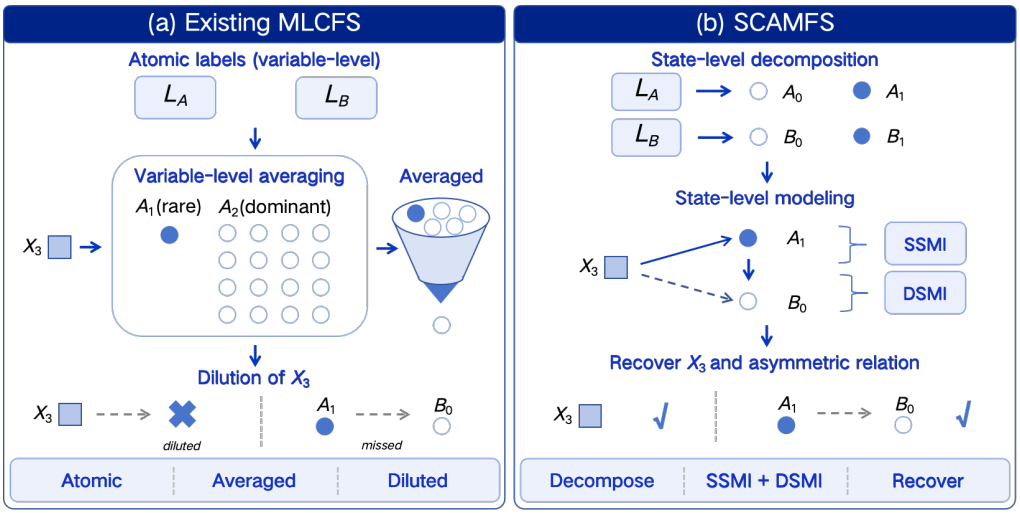


Figure 4: Motivation of SCAMFS. Existing MLCFS methods treat labels as atomic variables and average heterogeneous state-specific effects, so rare but important presence-state signals are overwhelmed by dominant absence-state noise, causing dilution of key features and missed asymmetric dependencies. SCAMFS addresses this by decomposing labels into explicit state nodes and using SSMI to measure the relevance between features and specific states, and DSMI to capture asymmetric cross-label dependencies, thereby recovering masked key features and revealing fine-grained state-level dependency structure.

Fig. 4 illustrates the motivation of SCAMFS by contrasting conventional variable-level label modeling with the proposed state-level modeling paradigm. Existing multi-label causal feature selection methods usually treat each binary label as an atomic variable, as shown in Fig. 4(a). Under this formulation, the presence and absence states of a label are compressed into a single statistical quantity. As a result, feature relevance is evaluated after variable-level averaging over different label states. This averaging can dilute state-specific signals. For example, when a feature is mainly informative for a rare present state, its contribution may be overwhelmed by the dominant absent state in the empirical distribution. Consequently, the feature may receive a weak global relevance score and be removed, even though it is useful for explaining a specific label state. This corresponds to the dilution of X_3 shown in Fig. 4(a).

Fig. 4(b) shows the state-level modeling idea adopted by SCAMFS. Each binary label is decomposed into two state-specific events, corresponding to its absence and presence states. Based on this decomposition, SCAMFS separately measures feature–state relevance and state–state dependency. SSMI is used to quantify whether a feature is informative for a specific label state, while DSMI is used to capture asymmetric dependencies between label states. This state-level view enables SCAMFS to distinguish heterogeneous presence/absence mechanisms and to identify asymmetric cross-label relations. It also helps alleviate label blocking: when a correlated label state masks a weak but informative feature, state-level explanatory competition allows SCAMFS to compare their conditional explanatory strengths and recover the feature if it provides stronger state-specific evidence.

B Detailed Algorithms of SCAMFS

B.1 Overall Four-Phase Procedure

Algorithm 1 summarizes the overall procedure of SCAMFS. The algorithm first decomposes labels into state-specific events and caches the unconditional SSMI and DSMI scores. It then processes each target label state through four phases: state-level cross-label dependency modeling, local state structure discovery, state-level feature recovery based on explanatory competition, and state-level marginal validity and redundancy elimination. The final feature subset is obtained by taking the union of the refined state-specific feature subsets over all target states.

Algorithm 1 SCAMFS: State-level Causal Asymmetry Modeling for Multi-label Feature Selection

Input: Data matrix $\mathcal{D}$, number of labels m, thresholds δ_1, δ_2, parameters k_1, k_2
Output: Globally selected feature subset $selfea$

```
Split D into feature set F and label set L = {L_1, ..., L_m}
Construct state events S = {E_{i,c} : E_{i,c} = I[L_i = c], L_i ∈ L, c ∈ Val(L_i)}
Cache all unconditional SSMI_c(X; L_i) and positive DSMI^+_{c,d}(L_i, L_j) scores
Initialize global_selected_features ← ∅
for each target state event E_{i,c} ∈ S do
    Phase 1: R_{i,c} ← State-level Cross-label Dependency Modeling (Alg. 2)
    Phase 2: CMB_{i,c} ← Local State Structure Discovery (Alg. 3)
    Phase 3: CMB_{i,c} ← State-level Feature Recovery Based on Explanatory Competition (Alg. 4)
    Phase 4: CMB_{i,c} ← State-level Marginal Validity and Cross-state Redundancy Elimination (Alg. 5)
    global_selected_features ← global_selected_features ∪ CMB_{i,c}
end for
return selfea = unique(global_selected_features)
```

B.2 Phase 1: State-level Cross-label Dependency Modeling

Algorithm 2 details the construction of the cross-label state dependency set $R_{i,c}$ for a target state $E_{i,c}$. SCAMFS first screens candidate label states using cached normalized DSMI^+ scores, then ranks the candidates and greedily retains only those whose dependency remains significant after conditioning on the already selected state events.

Algorithm 2 SCAMFS Phase 1: State-level Cross-label Dependency Modeling

Input: Target event $E_{i,c}$, label set $\mathcal{L}$, state-value sets $\{\mathrm{Val}(L_j)\}_{j=1}^{m}$, cached DSMI^+ scores, threshold δ_2, maximum conditioning size h
Output: Cross-label state dependency set $R_{i,c}$

```
Candidates ← {(E_{j,d}, score) : j ≠ i, d ∈ Val(L_j), E_{j,d} = I[L_j = d], score = DSMI^+_{c,d}(L_i, L_j), score > δ_2}
Sort Candidates by score in descending order
R_{i,c} ← ∅
for each (E_{j,d}, score) in Candidates do
```

if $R_{i,c} = \emptyset$ **then**
  $R_{i,c} \leftarrow R_{i,c} \cup \{E_{j,d}\}$
else
  $Z \leftarrow \mathrm{LimitConditioning}(R_{i,c}, h)$
  if $\mathrm{DSMI}^{+}_{c,d}(L_i, L_j \mid Z) > \delta_2$ **then**
    $R_{i,c} \leftarrow R_{i,c} \cup \{E_{j,d}\}$
  end if
end if
end for
return $R_{i,c}$

B.3 Phase 2: Local State Structure Discovery

Algorithm 3 details the local state structure discovery procedure. This phase first identifies direct state-relevant feature candidates by thresholding cached SSMI scores and applying the ResolveTopK rule controlled by k_1. It then performs conditional pruning under the current state context formed by the other PC candidates and $R_{i,c}$. Finally, SCAMFS searches for spouse candidates that are weak under $R_{i,c}$ alone but become informative after conditioning on a selected PC feature.

Algorithm 3 SCAMFS Phase 2: Local State Structure Discovery

Input: Target event $E_{i,c}$, feature set $\mathcal{F}$, dependency set $R_{i,c}$, cached SSMI scores, threshold δ_1, parameter k_1, spouse scan size s, maximum conditioning size h
Output: Local candidate Markov blanket $CMB_{i,c}$

$Cand_PC \leftarrow \{(X, \mathrm{SSMI}_c(X; L_i)) : X \in \mathcal{F},\ \mathrm{SSMI}_c(X; L_i) > \delta_1\}$
Sort $Cand_PC$ by SSMI score in descending order and keep the top $\mathrm{ResolveTopK}(|Cand_PC|, k_1)$ candidates
$PC_{\mathrm{indices}} \leftarrow$ feature indices in $Cand_PC$
for each $X \in PC_{\mathrm{indices}}$ **do**
  $Z \leftarrow \mathrm{LimitConditioning}((PC_{\mathrm{indices}} \setminus \{X\}) \cup R_{i,c}, h)$
  if $\mathrm{SSMI}_c(X; L_i \mid Z) \le \delta_1$ **then**
    $PC_{\mathrm{indices}} \leftarrow PC_{\mathrm{indices}} \setminus \{X\}$
  end if
end for
$SP_{\mathrm{indices}} \leftarrow \emptyset$
$Candidates_Z \leftarrow \mathcal{F} \setminus PC_{\mathrm{indices}}$
If $|Candidates_Z| > s$, keep the top s features ranked by cached $\mathrm{SSMI}_c(\cdot; L_i)$
for each $X \in PC_{\mathrm{indices}}$ **do**
  for each $Z_f \in Candidates_Z$ **do**
    $Z_R \leftarrow \mathrm{LimitConditioning}(R_{i,c}, h)$
    $Z_{XR} \leftarrow \mathrm{LimitConditioning}(\{X\} \cup R_{i,c}, h)$
    if $\mathrm{SSMI}_c(Z_f; L_i \mid Z_R) \le \delta_1$ **and** $\mathrm{SSMI}_c(Z_f; L_i \mid Z_{XR}) > \delta_1$ **then**
      $SP_{\mathrm{indices}} \leftarrow SP_{\mathrm{indices}} \cup \{Z_f\}$
    end if
  end for
end for
$CMB_{i,c} \leftarrow \mathrm{unique}(PC_{\mathrm{indices}} \cup SP_{\mathrm{indices}})$
return $CMB_{i,c}$

B.4 Phase 3: State-level Feature Recovery Based on Explanatory Competition

Algorithm 4 details the recovery procedure based on explanatory competition. This phase revisits missing features that were not selected in Phase 2 but still have weak state-specific relevance. For each such feature, SCAMFS compares its conditional explanatory strength with that of each active cross-label state. If the feature provides stronger conditional evidence, it is recovered into $CMB_{i,c}$ and the corresponding blocking state is deactivated. The implementation follows a single-success strategy and stops after the first successful replacement for each target state.

Algorithm 4 SCAMFS Phase 3: State-level Feature Recovery Based on Explanatory Competition

Input: Target event $E_{i,c}$, feature set $\mathcal{F}$, current candidate blanket $CMB_{i,c}$, PC set PC_{indices}, dependency set $R_{i,c}$, cached SSMI scores, threshold δ_1, scan size s, maximum conditioning size h

Output: Updated candidate blanket $CMB_{i,c}$

$F_{\text{miss}} \leftarrow \mathcal{F} \setminus CMB_{i,c}$
$F_{\text{miss}} \leftarrow \{X \in F_{\text{miss}} : \text{SSMI}_c(X; L_i) \geq \delta_1/2\}$
If $|F_{\text{miss}}| > s$, keep the top s features ranked by cached $\text{SSMI}_c(\cdot; L_i)$
Mark all states in $R_{i,c}$ as active
for each $X \in F_{\text{miss}}$ **do**
  for each active state $E_{j,d} \in R_{i,c}$ **do**
    $Z_{\text{base}} \leftarrow \text{LimitConditioning}(PC_{\text{indices}} \cup (R_{i,c} \setminus \{E_{j,d}\}), h)$
    $score_X \leftarrow \text{SSMI}_c(X; L_i \mid Z_{\text{base}})$
    $score_B \leftarrow \text{DSMI}^+_{c,d}(L_i, L_j \mid Z_{\text{base}})$
    if $score_X > score_B$ **then**
      $CMB_{i,c} \leftarrow CMB_{i,c} \cup \{X\}$
      Deactivate $E_{j,d}$ in $R_{i,c}$
      break
    end if
  end for
  if X has been recovered **then**
    break
  end if
end for
return $CMB_{i,c}$

B.5 Phase 4: State-level Marginal Validity and Redundancy Elimination

Algorithm 5 details the final refinement phase of SCAMFS. This phase first ranks the current candidate blanket by state-specific relevance, keeps the top candidates according to k_2, then removes features that fail marginal validity or are more strongly associated with other label states.

Algorithm 5 SCAMFS Phase 4: State-level Marginal Validity and Redundancy Elimination

Input: Target state $L_i = c$, candidate blanket $CMB_{i,c}$, label set $\mathcal{L}$, state-value sets $\{\text{Val}(L_j)\}_{j=1}^{m}$, cached SSMI scores, threshold δ_1, parameter k_2

Output: Refined feature set $F_{i,c}^{\text{final}}$

Rank features in $CMB_{i,c}$ by $\text{SSMI}_c(X; L_i)$ in descending order
$TopCMB \leftarrow$ top $\text{ResolveTopK}(|CMB_{i,c}|, k_2)$ features
$F_{i,c}^{\text{valid}} \leftarrow \{X \in TopCMB_{i,c} : \text{SSMI}_c(X; L_i) > \delta_1\}$
$F_{i,c}^{\text{final}} \leftarrow F_{i,c}^{\text{valid}}$
for each feature $X \in F_{i,c}^{\text{valid}}$ **do**
  for each label $L_j \in \mathcal{L} \setminus \{L_i\}$ **do**
    for each state value $d \in \text{Val}(L_j)$ **do**
      if $\text{SSMI}_d(X; L_j) > 1.2 \times \text{SSMI}_c(X; L_i)$ **then**
        $F_{i,c}^{\text{final}} \leftarrow F_{i,c}^{\text{final}} \setminus \{X\}$
        break
      end if
    end for
  end for
end for
return $F_{i,c}^{\text{final}}$

C Derivations and Discussions of State-level Information Metrics

C.1 Derivation of Variable-level Mutual Information Decomposition

For a discrete feature variable X and a binary label variable $L_j \in \{0, 1\}$, the mutual information between X and L_j is defined as

$$I(X; L_j) = \sum_{x} \sum_{c \in \{0,1\}} P(x, L_j = c) \log \frac{P(x, L_j = c)}{P(x)P(L_j = c)}. \tag{19}$$

Using the identity $P(x, L_j = c) = P(L_j = c)P(x \mid L_j = c)$, we have

$$\begin{aligned} I(X; L_j) &= \sum_{c \in \{0,1\}} \sum_{x} P(L_j = c)P(x \mid L_j = c) \log \frac{P(L_j = c)P(x \mid L_j = c)}{P(x)P(L_j = c)} \\ &= \sum_{c \in \{0,1\}} P(L_j = c) \sum_{x} P(x \mid L_j = c) \log \frac{P(x \mid L_j = c)}{P(x)} \\ &= \sum_{c \in \{0,1\}} P(L_j = c) D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big). \end{aligned} \tag{20}$$

This proves Eq. (1). The decomposition shows that variable-level mutual information aggregates the distributional deviation of each label state with weight $P(L_j = c)$. Therefore, the contribution of a rare but informative state can be down-weighted in the global mutual information score. The following propositions further clarify why this weighted aggregation motivates state-level information scores.

C.2 Why State-level Information Scores Are Needed

The variable-level mutual information decomposition in Appendix C.1 shows that different label states contribute to $I(X; L_j)$ with different empirical weights. This section further formalizes two consequences of this observation: rare state-specific signals may be diluted by variable-level aggregation, and ordinary mutual information with complementary state indicators cannot distinguish presence-specific and absence-specific relevance.

Proposition 1 (State-level dilution under variable-level mutual information)**.** *For a binary label $L_j \in \{0, 1\}$, let*

$$a_c = D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big).$$

Then the variable-level mutual information can be written as

$$I(X; L_j) = P(L_j = 0)a_0 + P(L_j = 1)a_1.$$

Consequently, if a rare state $L_j = c$ has probability $p_c = P(L_j = c)$, its contribution to the variable-level score is $p_c a_c$, even when the state-specific deviation a_c is large.

Proof. The result follows directly from the mutual information decomposition:

$$I(X; L_j) = \sum_{c \in \{0,1\}} P(L_j = c) D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big).$$

Thus, the contribution of state c is weighted by $P(L_j = c)$. When $P(L_j = c)$ is small, a large state-specific deviation may still make only a small contribution to the global variable-level score. □

Proposition 2 (Invariance of mutual information under complementary state indicators)**.** *Let $E_{j,1} = \mathbb{I}[L_j = 1]$ and $E_{j,0} = \mathbb{I}[L_j = 0]$. Since $E_{j,0} = 1 - E_{j,1}$ is a bijective transformation of $E_{j,1}$, ordinary mutual information satisfies*

$$I(X; E_{j,0}) = I(X; E_{j,1}).$$

Therefore, ordinary mutual information with binary state indicators cannot distinguish whether X is more informative for the presence state or the absence state.

Proof. Mutual information is invariant under any bijective transformation of one random variable. Since $E_{j,0} = 1 - E_{j,1}$ defines a one-to-one transformation between the two binary indicators, we have

$$I(X; E_{j,0}) = I(X; E_{j,1}).$$

Thus, ordinary mutual information with state indicators gives identical scores for the two complementary states. □

Proposition 3 (State-specific separability of SSMI)**.** *For a binary label L_j, the two SSMI scores*

$$\mathrm{SSMI}_0(X; L_j) = D_{\mathrm{KL}}\big(P(X \mid L_j = 0) \| P(X)\big)$$

and

$$\mathrm{SSMI}_1(X; L_j) = D_{\mathrm{KL}}\big(P(X \mid L_j = 1) \| P(X)\big)$$

need not be equal. In particular, whenever the two conditional feature distributions induce different deviations from the marginal distribution $P(X)$, i.e.,

$$D_{\mathrm{KL}}\big(P(X \mid L_j = 0) \| P(X)\big) \neq D_{\mathrm{KL}}\big(P(X \mid L_j = 1) \| P(X)\big),$$

the two state-specific SSMI scores are different. Therefore, SSMI can distinguish feature relevance to different label states, unlike ordinary mutual information with complementary state indicators.

Proof. SSMI evaluates the KL divergence between the feature distribution under a specific label state and the marginal feature distribution. Since $P(X \mid L_j = 0)$ and $P(X \mid L_j = 1)$ may deviate from $P(X)$ by different amounts, their KL divergences to $P(X)$ may also differ. Hence, the two state-specific SSMI scores need not be equal. □

C.3 Conditional Extension of SSMI

The unconditional SSMI in Eq. (4) measures the distributional deviation between the feature distribution under a target state and the marginal feature distribution:

$$\mathrm{SSMI}_c(X; L_j) = D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big).$$

When a conditioning set Z is given, this comparison should be performed within each configuration $Z = z$, because the conditioning variables may explain part of the feature–state association. Therefore, SCAMFS uses the following conditional extension:

$$\mathrm{SSMI}_c(X; L_j \mid Z) = \sum_z P(z \mid L_j = c) D_{\mathrm{KL}}\big(P(X \mid Z = z, L_j = c) \| P(X \mid Z = z)\big).$$

This quantity averages the state-specific distributional deviation over the configurations of Z that occur under the target state $L_j = c$. The weight $P(z \mid L_j = c)$ emphasizes the conditional contexts that are actually observed within the target state. Thus, the conditional SSMI measures whether X still provides state-specific information after the effect of Z has been taken into account.

C.4 Conditional Extension of DSMI

The unconditional DSMI in Eq. (6) measures a directional association from a source state $L_i = c$ to a target state $L_j = d$:

$$\mathrm{DSMI}_{c,d}(L_i, L_j) = P(L_j = d \mid L_i = c) \log \frac{P(L_j = d \mid L_i = c)}{P(L_j = d)}.$$

To account for a conditioning set Z, SCAMFS evaluates this directional association within each conditional context $Z = z$ and then averages over the contexts observed under the source state $L_i = c$:

$$\mathrm{DSMI}_{c,d}(L_i, L_j \mid Z) = \sum_z P(z \mid L_i = c) P(L_j = d \mid L_i = c, z) \log \frac{P(L_j = d \mid L_i = c, z)}{P(L_j = d \mid z)}.$$

Here, $P(z \mid L_i = c)$ assigns larger weight to conditioning configurations that are more likely under the source state. The ratio inside the logarithm compares the probability of the target state $L_j = d$ with and without the source state $L_i = c$ under the same conditioning context. Therefore, the conditional DSMI measures whether the source state still provides directional evidence for the target state after controlling for Z.

C.5 Normalization by State Surprise

In implementation, SCAMFS normalizes the raw SSMI and DSMI scores by the surprise of the corresponding target state. Specifically, the feature–state score is normalized by $-\log P(L_j = c)$, and the state–state score is normalized by $-\log P(L_j = d)$. The normalized forms are:

$$\widetilde{\mathrm{SSMI}}_c(X; L_j) = \frac{D_{\mathrm{KL}}\big(P(X \mid L_j = c) \| P(X)\big)}{-\log P(L_j = c)},$$

and

$$\widetilde{\mathrm{DSMI}}^{+}_{c,d}(L_i, L_j) = \frac{\max\big(0, \mathrm{DSMI}_{c,d}(L_i, L_j)\big)}{-\log P(L_j = d)}.$$

This normalization is motivated by the fact that label states can have very different empirical frequencies in multi-label data. Raw state-specific scores associated with rare states and frequent states may not be directly comparable. The term $-\log P(\cdot)$ corresponds to the surprise of observing a state; dividing by this quantity rescales the score according to the rarity of the target state. This makes scores across different states more comparable. The normalization is especially useful in the explanatory competition phase of SCAMFS. In that phase, a missing feature competes with a blocking label state, and the algorithm compares a feature–state score with a state–state score under the same target state. Normalizing SSMI and DSMI by state surprise places these quantities on a more compatible scale, making the competition more stable across labels with different state frequencies.

C.6 Auxiliary Definitions for Local State Structure Discovery

Definition 4 (State-level Spouse Candidate)**.** *For a target state $L_i = c$ and its cross-label state context $R_{i,c}$, a feature $Z \notin PC_{\mathrm{indices}}$ is regarded as a state-level spouse candidate if there exists a feature such that*

$$\mathrm{SSMI}_c(Z; L_i \mid R_{i,c}) \le \delta_1 \quad \textit{and} \quad \mathrm{SSMI}_c(Z; L_i \mid \{X\} \cup R_{i,c}) > \delta_1.$$

This condition indicates that Z is weakly associated with the target state under the current cross-label state context, but becomes informative after conditioning on a selected PC feature. It captures a state-level interaction pattern analogous to a V-structure without requiring full graph recovery.

Definition 5 (State-specific Candidate Markov Blanket)**.** *For a target state $L_i = c$, the state-specific candidate Markov blanket is defined as*

$$CMB_{i,c} = PC_{\mathrm{indices}} \cup SP_{\mathrm{indices}},$$

where PC_{indices} denotes the features that remain state-relevant after conditional PC pruning, and SP_{indices} denotes the state-level spouse candidates. The set $CMB_{i,c}$ is a candidate local feature structure for the target state and is further refined by causal recovery, marginal validity checking, and cross-state redundancy elimination.

C.7 Auxiliary Definitions and Recovery Criterion for Phase 3

Definition 6 (State-level Blocking State)**.** *For a target state $L_i = c$, a selected cross-label state $E_{j,d} \in R_{i,c}$ is called a blocking state for a feature $X \in \mathcal{F} \setminus CMB_{i,c}$ if the state $E_{j,d}$ accounts for part of the state-specific association between X and $L_i = c$ under the current local context. In this case, X may be excluded from the local candidate Markov blanket although it still contains useful state-specific information.*

Definition 7 (Explanatory Competition)**.** *Given a missing feature $X \in \mathcal{F} \setminus CMB_{i,c}$ and a blocking state $E_{j,d} \in R_{i,c}$, SCAMFS constructs the base conditioning set*

$$Z_{\mathrm{base}} = PC_{\mathrm{indices}} \cup (R_{i,c} \setminus \{E_{j,d}\}).$$

The explanatory competition compares the conditional feature–state score

$$\mathrm{SSMI}_c(X; L_i \mid Z_{\mathrm{base}})$$

with the conditional state–state score

$$\mathrm{DSMI}^{+}_{c,d}(L_i, L_j \mid Z_{\mathrm{base}}).$$

Criterion 1 (State-level Feature Reassessment Criterion)**.** *For a target state $L_i = c$, let $X \in \mathcal{F} \setminus CMB_{i,c}$ be a missing feature satisfying*

$$\mathrm{SSMI}_c(X; L_i) \geq \delta_1/2.$$

If there exists a selected label state $E_{j,d} \in R_{i,c}$ such that

$$\mathrm{SSMI}_c(X; L_i \mid Z_{\mathrm{base}}) > \mathrm{DSMI}^+_{c,d}(L_i, L_j \mid Z_{\mathrm{base}}),$$

where

$$Z_{\mathrm{base}} = PC_{\mathrm{indices}} \cup (R_{i,c} \setminus \{E_{j,d}\}),$$

then SCAMFS reassesses X as informative for the target state, adds X into $CMB_{i,c}$, and deactivates the competing blocking state $E_{j,d}$ from the active dependency set.

Justification. The reassessment phase first restricts the search space to missing features with weak but non-negligible marginal state relevance, namely $\mathrm{SSMI}_c(X; L_i) \geq \delta_1/2$. For each such feature X and each active label state $E_{j,d} \in R_{i,c}$, SCAMFS evaluates both quantities under the same base conditioning set Z_{base}. If

$$\mathrm{SSMI}_c(X; L_i \mid Z_{\mathrm{base}}) > \mathrm{DSMI}^+_{c,d}(L_i, L_j \mid Z_{\mathrm{base}}),$$

then X provides stronger conditional evidence for the target state than the competing label state under the same context. According to the Phase 3 reassessment rule, SCAMFS adds X into $CMB_{i,c}$ and removes $E_{j,d}$ from the active dependency set. Therefore, the stated inequality is used as the state-level feature reassessment criterion in Phase 3. □

C.8 Auxiliary Definitions and Retention Criterion for Phase 4

Definition 8 (State-level Marginal Validity)**.** *For a target state $L_i = c$, a feature X is said to satisfy state-level marginal validity if*

$$\mathrm{SSMI}_c(X; L_i) > \delta_1.$$

This condition ensures that X still has sufficient marginal state-specific relevance to the target state after the recovery phase.

Definition 9 (Cross-state Redundant Feature)**.** *For a target state $L_i = c$, a retained feature X is considered cross-state redundant if there exists another label state $L_j = d$ with $j \neq i$ such that*

$$\mathrm{SSMI}_d(X; L_j) > 1.2 \times \mathrm{SSMI}_c(X; L_i).$$

In this case, X is more strongly associated with a non-target state than with the current target state and is removed from $CMB_{i,c}$.

Criterion 2 (Phase-4 Feature Retention Criterion)**.** *For a target state $L_i = c$, a feature $X \in CMB_{i,c}$ is retained after Phase 4 if it satisfies all of the following conditions:*

$$X \in \mathrm{TopK}_{k_2}(CMB_{i,c}),$$

$$\mathrm{SSMI}_c(X; L_i) > \delta_1,$$

and for every non-target state $L_j = d$ with $j \neq i$,

$$\mathrm{SSMI}_d(X; L_j) \leq 1.2 \times \mathrm{SSMI}_c(X; L_i).$$

Justification. Phase 4 first ranks all features in $CMB_{i,c}$ by their cached state-specific relevance scores $\mathrm{SSMI}_c(X; L_i)$ and keeps only the top $\mathrm{ResolveTopK}(|CMB_{i,c}|, k_2)$ features. Therefore, a feature must belong to this top-ranked subset to be considered further. Next, SCAMFS checks marginal validity and retains a feature only if

$$\mathrm{SSMI}_c(X; L_i) > \delta_1.$$

Any feature failing this condition is removed. Finally, SCAMFS performs cross-state redundancy elimination. For each remaining feature X, it compares its relevance to the target state $L_i = c$ with its relevance to every non-target state $L_j = d$, where $j \neq i$. If there exists a non-target state satisfying

$$\mathrm{SSMI}_d(X; L_j) > 1.2 \times \mathrm{SSMI}_c(X; L_i),$$

then X is removed as cross-state redundant. Hence, if no such non-target state exists, and the previous top-k_2 and marginal validity conditions hold, X is retained after Phase 4. □

D Complexity Analysis

Let N be the number of samples, d the number of features, and q the number of decomposed label states, where $q = \sum_{j=1}^{m} |\mathrm{Val}(L_j)|$. For binary labels, $q = 2m = 2|\mathcal{L}|$. Let h denote the maximum conditioning-set size used by $\mathrm{LimitConditioning}(\cdot)$, and let $s = \min(d, \max(20, \lceil 0.25d \rceil))$ denote the maximum number of features scanned during spouse discovery and recovery.

SCAMFS first caches all unconditional feature–state and state–state scores. Computing the SSMI cache requires evaluating dq feature–state pairs, and computing the DSMI cache requires evaluating at most q^2 state–state pairs. Since each empirical information score is computed over N samples, the caching cost is $O(Ndq + Nq^2)$. This cache avoids repeated unconditional score computation in later phases.

For each target state, Phase 1 scans the remaining label states and sorts the candidates by cached DSMI^+ scores, which costs $O(q \log q)$. The greedy conditional filtering step performs conditional DSMI tests on selected candidates. If $r_R = |R_{i,c}|$ denotes the number of retained cross-label states, this step costs at most $O(Nhr_R)$, where h is bounded by a small constant in the implementation.

Phase 2 first screens all features using cached SSMI scores and retains the top $\mathrm{ResolveTopK}(|Cand_PC|, k_1)$ candidates. Let r_1 be the number of retained PC candidates. Conditional PC pruning costs $O(Nhr_1)$. Spouse discovery then scans at most s non-PC candidate features for each retained PC feature, yielding $O(Nr_1 s)$, when h is treated as a constant. This is the dominant cost in practice.

Phase 3 considers only missing features with $\mathrm{SSMI}_c(X; L_i) \geq \delta_1/2$ and further caps the scan size by s. For each remaining candidate, it compares the conditional SSMI of the feature with the conditional DSMI of active blocking states in $R_{i,c}$. Thus, the worst-case cost per target state is $O(Nhs|R_{i,c}|)$, or $O(Ns|R_{i,c}|)$ for constant h. The single-success recovery strategy often makes the practical cost lower because the loop stops after the first successful feature–state replacement.

Phase 4 sorts the current candidate set $CMB_{i,c}$ by cached SSMI scores, keeps the top $\mathrm{ResolveTopK}(|CMB_{i,c}|, k_2)$ features, and checks marginal validity and cross-state redundancy. Let $r_2 = |CMB_{i,c}|$ after recovery. The sorting cost is $O(r_2 \log r_2)$, and the redundancy check costs $O(r_2 q)$ using cached SSMI scores.

Combining all target states, the worst-case complexity is dominated by Phase 2. Since $r_1 \leq d$ and $s \leq d$, the worst-case cost is bounded by $O(qNd^2)$, in addition to the initial cache construction cost $O(Ndq + Nq^2)$.

In practice, the cost is substantially lower because SCAMFS uses cached unconditional scores, top-k candidate truncation, bounded conditioning sets, capped spouse/recovery scans, and single-success recovery.

E Dataset Statistics

Table 3: Description of Datasets

Datasets	Domains	Instances	Features	Labels	Cardinality	Density	AvgIR
Flags	Image	194	19	7	3.392	0.485	2.255
VirusGO	Biology	207	749	6	1.217	0.203	4.041
CHD_49	Medicine	555	49	6	2.58	0.43	5.766
PlantGO	Biology	978	3091	12	1.079	0.090	6.690
Enron	Text	1702	1001	53	3.378	0.064	73.953
Image	Image	2000	294	5	1.236	0.247	1.193
Yeast	Biology	2417	103	14	4.237	0.303	7.197

F Evaluation Metrics for Multi-label Classification

This section provides the formal definitions and brief interpretations of the seven evaluation metrics used to evaluate SCAMFS. Let $\mathcal{D} = \{(x_i, Y_i)\}_{i=1}^{N}$ be the test set, where $Y_i \subseteq \mathcal{L}$ denotes the set of relevant labels for instance x_i, and $h(x_i) \subseteq \mathcal{L}$ denotes the predicted label set. Let $f(x_i, l)$ be the confidence score for label l, and let $\mathrm{rank}_i(l)$ denote the rank of label l in descending order of $f(x_i, \cdot)$. For vector-based metrics, Y_i and $h(x_i)$ are represented by their corresponding binary indicator vectors.

F.1 Hamming Loss (HL)

Hamming Loss evaluates the average fraction of instance-label pairs that are misclassified:

$$HL = \frac{1}{N \cdot L} \sum_{i=1}^{N} |h(x_i) \Delta Y_i|, \tag{21}$$

where Δ denotes the symmetric difference between the predicted and ground-truth label sets. A smaller value indicates better performance.

F.2 Subset Accuracy (SA)

Subset Accuracy is the most stringent metric in multi-label learning, as it requires the predicted label set to exactly match the ground-truth label set:

$$SA = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big(h(x_i) = Y_i\big), \tag{22}$$

where $\mathbb{I}(\cdot)$ is the indicator function. A larger value indicates better performance.

F.3 Average Precision (AvP)

Average Precision evaluates how well relevant labels are ranked ahead of irrelevant labels:

$$AvP = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|Y_i|} \sum_{l \in Y_i} \frac{|\{l' \in Y_i : \mathrm{rank}_i(l') \leq \mathrm{rank}_i(l)\}|}{\mathrm{rank}_i(l)}. \tag{23}$$

A larger value indicates better ranking performance.

F.4 Coverage (Cov)

Coverage measures how far one needs to go down the ranked label list to cover all relevant labels for each instance:

$$Cov = \frac{1}{N} \sum_{i=1}^{N} \max_{l \in Y_i} \mathrm{rank}_i(l) - 1. \tag{24}$$

A smaller value indicates that relevant labels are ranked closer to the top.

F.5 Ranking Loss (RL)

Ranking Loss measures the proportion of relevant–irrelevant label pairs that are incorrectly ordered:

$$RL = \frac{1}{N} \sum_{i=1}^{N} \frac{|\{(l_a, l_b) : f(x_i, l_a) \leq f(x_i, l_b), l_a \in Y_i,\ l_b \notin Y_i\}|}{|Y_i|\,|\bar{Y}_i|}, \tag{25}$$

where $\bar{Y}_i = \mathcal{L} \setminus Y_i$ denotes the set of irrelevant labels. A smaller value indicates better ranking performance.

F.6 MACRO-F1 (FMA)

Macro-F1 computes the arithmetic mean of the F1 scores calculated independently for each label:

$$Fma = \frac{1}{L}\sum_{j=1}^{L}\frac{2TP_j}{2TP_j + FP_j + FN_j}, \tag{26}$$

where TP_j, FP_j, and FN_j denote the true positives, false positives, and false negatives for the j-th label. A larger value indicates better performance.

F.7 MICRO-F1 (FMI)

Micro-F1 computes the F1 score globally by aggregating true positives, false positives, and false negatives over all labels:

$$Fmi = \frac{2\sum_{j=1}^{L} TP_j}{2\sum_{j=1}^{L} TP_j + \sum_{j=1}^{L} FP_j + \sum_{j=1}^{L} FN_j}. \tag{27}$$

A larger value indicates better overall performance and is more sensitive to frequent labels than Macro-F1.

G SUPPLEMENTARY RESULTS FOR STATE-OF-THE-ART COMPARISON

This section reports the supplementary results for the comparison between SCAMFS and seven representative multi-label feature selection methods. The main paper presents *Hamming Loss* and *Micro-F1*; here we provide the remaining five metrics: *Subset Accuracy*, *Average Precision*, *Coverage*, *Ranking Loss*, and *Macro-F1*. Best results are marked in bold, and second-best results are underlined.

G.1 SUBSET ACCURACY

Table 4 reports the Subset Accuracy results. SCAMFS achieves the best performance on all seven datasets, indicating that the selected features improve exact-match multi-label prediction.

Table 4: SUBSET ACCURACY FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	SUBSET ACCURACY (↑)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.2105**	**0.8158**	**0.1786**	**0.8041**	**0.2468**	**0.5250**	**0.2479**
MI-MCF	<u>0.1687</u>	0.6452	<u>0.1687</u>	0.5870	0.2427	0.4783	<u>0.2097</u>
M2LC	0.0517	0.4677	0.1506	0.6587	0.2377	0.3583	0.1738
D2F	0.0517	0.6936	0.1506	0.6553	0.2388	0.4650	0.1628
MDFS	0.0862	0.5714	0.1566	0.5870	0.2407	<u>0.4900</u>	0.1931
JFSC	0.1552	0.6936	0.1024	<u>0.6918</u>	<u>0.2466</u>	0.4600	0.1986
FIMF	0.1207	<u>0.7049</u>	0.1084	0.6041	0.2407	0.4817	0.1986
MCLS	0.1035	0.5968	0.1265	0.6464	0.2232	0.4833	0.1959

G.2 AVERAGE PRECISION

Table 5 reports the Average Precision results. SCAMFS obtains the best performance on most datasets and remains competitive on *Flags*, showing strong label-ranking ability.

Table 5: AVERAGE PRECISION FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	AVERAGE PRECISION (↑)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	0.7824	**0.9529**	**0.8145**	**0.8951**	**0.3778**	**0.7969**	**0.7884**
MI-MCF	**0.7907**	0.9356	0.8066	0.8389	0.3710	0.7699	0.7650
M2LC	0.7815	0.8979	0.7622	0.8545	0.3380	0.7502	0.7548
D2F	0.7318	0.9164	0.7752	0.8788	0.2829	0.7683	0.7511
MDFS	0.7626	0.8949	0.7585	0.8651	0.2281	0.7894	0.7552
JFSC	0.7882	0.9086	0.7846	0.8651	0.3226	0.7468	0.7524
FIMF	0.7518	0.9486	0.7710	0.8698	0.2810	0.7832	0.7573
MCLS	0.7826	0.8298	0.8001	0.8572	0.3054	0.7647	0.7590

G.3 COVERAGE

Table 6 reports the Coverage results. Lower values indicate better performance. SCAMFS achieves strong results on most datasets, suggesting that relevant labels tend to appear near the top of the ranked prediction list.

Table 6: COVERAGE FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	COVERAGE (↓)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.5263**	**0.0526**	**0.5214**	**0.0553**	0.2916	0.2325	**0.4463**
MI-MCF	0.6581	0.0871	0.5277	0.0555	0.2848	0.2663	0.4862
M2LC	0.6839	0.0871	0.5602	0.0565	**0.2808**	0.2800	0.4914
D2F	0.6121	0.1161	0.5482	0.0568	0.3440	0.2604	0.5021
MDFS	0.6523	0.1071	0.5831	0.0633	0.3579	0.2471	0.4875
JFSC	0.6552	0.0968	0.6552	0.0635	0.3131	0.2737	0.4872
FIMF	0.6667	0.1082	0.5976	0.0602	0.3201	**0.2068**	0.4927
MCLS	0.6351	0.1839	0.5410	0.0802	0.3252	0.2592	0.4800

G.4 RANKING LOSS

Table 7 reports the Ranking Loss results. SCAMFS achieves the lowest or near-lowest ranking loss on most datasets, indicating reliable ordering between relevant and irrelevant labels

Table 7: RANKING LOSS FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	RANKING LOSS (↓)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.1816**	**0.0257**	**0.1768**	0.0468	**0.1435**	**0.1696**	**0.1409**
MI-MCF	0.2486	0.0399	0.1971	0.0470	0.1447	0.1985	0.1688
M2LC	0.2540	0.0440	0.2293	0.0477	0.1762	0.2164	0.1717
D2F	0.2529	0.0556	0.2292	0.0471	0.1740	0.1928	0.1784
MDFS	0.2586	0.0661	0.2421	0.0468	0.1911	0.1786	0.1713
JFSC	0.2239	0.0472	0.2182	0.0542	0.1509	0.2100	0.1746
FIMF	0.2528	0.0324	0.2334	**0.0456**	0.1662	0.1888	0.1735
MCLS	0.2443	0.1266	0.1999	0.0717	0.1656	0.1960	0.1695

G.5 MACRO-F1

Table 8 reports the Macro-F1 results. SCAMFS performs strongly across datasets, showing that the selected features are useful for both frequent and less frequent labels.

Table 8: MACRO-F1 FOR EIGHT METHODS ON SEVEN DATASETS

METHOD	MACRO-F1 (↑)						
	FLAGS	VIRUSGO	CHD_49	PLANTGO	ENRON	IMAGE	YEAST
SCAMFS	**0.6983**	**0.7431**	0.4722	**0.6639**	**0.6484**	**0.6237**	**0.4114**
MI-MCF	0.5533	0.5513	**0.4946**	0.5076	0.4902	0.5928	0.4042
M2LC	0.5270	0.5266	0.4542	0.6287	0.5715	0.5203	0.3746
D2F	0.5329	0.6787	0.4554	0.5693	0.5428	0.5783	0.3590
MDFS	0.3635	0.5672	0.4375	0.5525	0.5240	0.6132	0.3780
JFSC	0.6176	0.5773	0.4186	0.5808	0.6037	0.5642	0.3807
FIMF	0.5155	0.5744	0.3982	0.5530	0.4985	0.5895	0.3874
MCLS	0.4094	0.5159	0.4444	0.5703	0.4715	0.5807	0.3926

H NORMALIZED RADAR VISUALIZATION

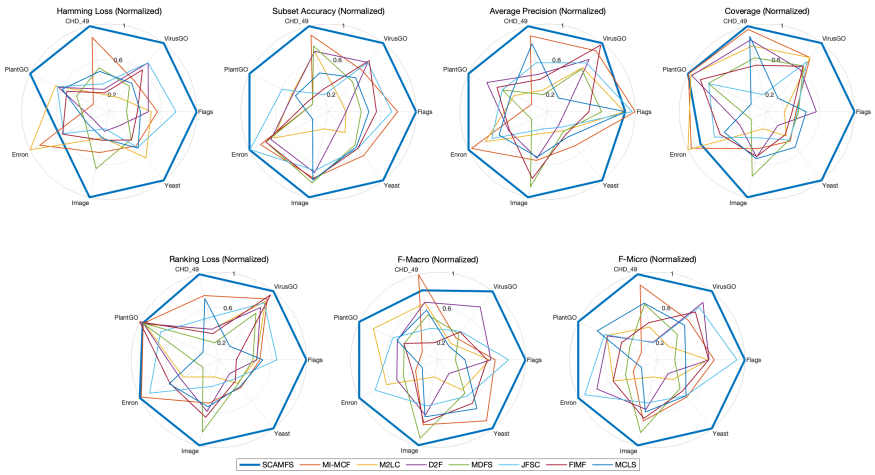


Figure 5: Radar charts of eight multi-label feature selection methods across seven metrics on seven datasets. The bold blue line represents the proposed SCAMFS method.

I STATISTICAL SIGNIFICANCE ANALYSIS

Table 9: SUMMARY OF THE FRIEDMAN STATISTICS F_F IN TERMS OF EACH EVALUATION METRIC AND CRITICAL VALUE

Evaluation metric	F_F	Critical value $\alpha = 0.05$
Hamming Loss	3.3670	2.2371
Subset Accuracy	5.1464	
Average Precision	3.6834	
Coverage	3.6987	
Ranking Loss	4.1124	
Macro-F1	3.4548	
Micro-F1	4.2900	

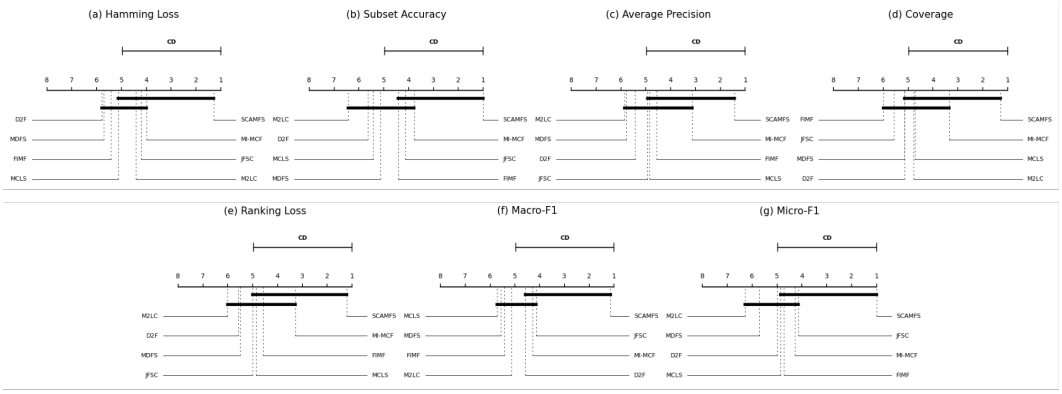


Figure 6: Comparison of SCAMFS against its rivals with the Bonferroni–Dunn test at a significance level $\alpha = 0.05$. (a) Hamming Loss. (b) Subset Accuracy. (c) Average Precision. (d) Coverage. (e) Ranking Loss. (f) Macro-F1. (g) Micro-F1.

J RUN-TIME COMPARISON

Table 10 reports the run-time comparison of SCAMFS and the seven baseline methods on the evaluated datasets. The results show that SCAMFS remains computationally practical while performing state-level dependency modeling.

Table 10: Run-time Comparison (Seconds) on Different Datasets.

Datasets	SCAMFS	MI-MCF	M2LC	D2F	MDFS	JFSC	FIMF	MCLS
Flags	**0.14**	0.83	0.56	0.52	0.50	0.36	0.34	0.40
VirusGO	0.43	59.24	0.82	3.35	1.65	0.41	**0.33**	0.41
CHD_49	**0.12**	0.57	0.39	0.55	0.42	0.38	0.31	0.47
PlantGO	44.18	186.02	26.74	151.29	23.19	4.25	0.43	**0.41**
Enron	25.60	4038.30	475.34	33.57	3.42	1.20	**0.76**	17.24
Image	20.80	7.59	1.04	32.41	3.30	**0.68**	2.51	1.69
Yeast	8.05	10.26	**2.02**	6.47	8.63	8.65	6.59	4.39