Towards Practical Multi-label Causal Discovery in High-Dimensional Event Sequences via One-Shot Graph Aggregation

Hugo Math[1,2] **Rainer Lienhart**[2]

[1] BMW Group
[2] Chair for Machine Learning and Computer Vision, Augsburg University
Augsburg, Germany
hugo.math@bmwgroup.de, rainer.lienhart@uni-augsburg.de

Abstract

Understanding causality in event sequences where outcome labels such as diseases or system failures arise from preceding events like symptoms or error codes is critical—yet remains an unsolved challenge across domains like healthcare or vehicle diagnostics. We introduce CARGO, a scalable multi-label causal discovery method for sparse, high-dimensional event sequences consisting of thousands of unique event types. Using two pretrained causal Transformers as domain-specific foundation models for event sequences—CARGO infers in parallel, per sequence, one-shot causal graphs and aggregates them using an adaptive frequency fusion to reconstruct the global Markov boundaries of labels. This two-stage approach enables efficient probabilistic reasoning at scale while avoiding the intractable cost of full-dataset conditional-independence testing. Our results on a challenging real-world automotive fault prediction dataset with over 29,100 unique event types and 474 imbalanced labels demonstrate CARGO's ability to perform structured reasoning.

1 Introduction

Understanding *why* specific events lead to particular outcomes is vital for effective diagnosis, predictions, and overall decision-making [24, 36]. For instance, "what series of events captured by diagnostic led to this vehicle failure" or "what symptoms led to this disease" [27, 39, 15, 25, 34]. Here, an event sequence consists of a list of discrete events x_i recorded asynchronously over time, while labels $\boldsymbol{y}$ summarize outcomes associated with the full sequence (e.g, a diagnosed defect or condition).

A fundamental obstacle in these settings is dimensionality. Real-world systems often involve tens of thousands of possible events, rendering causal discovery intractable for current algorithms [14]. To address this, we reinterpret multi-label causal discovery for event sequences as a form of Bayesian model averaging [16, 33], where each sequence is treated as a sample from a local causal model. Specifically, each sequence induces a one-shot causal graph (i.e., a directed acyclic graph (DAG)) [40]. Together, they can be fused to form a unified global structure [6]. This process, known as structural fusion [32], aggregates local graphs into a consensus causal graph over all observed sequences.

To summarize our contributions: we introduce CARGO (Causal Aggregation via Regressive Graph Operations), the first method to provide scalable causal discovery across thousands of labelled event sequences, with theoretical guarantees under standard assumptions. It is divided into two phases: (1)

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Structured Probabilistic Inference and Generative Modeling.

One-shot graph extraction, where for each sequence, CARGO infers the local Markov boundary of each label using two autoregressive Transformers as density estimators [48, 10] (2) Graph fusion, where the local graphs are aggregated via an adaptive threshold function to provide global Markov Boundaries.

We empirically validate CARGO on a large-scale vehicular dataset comprising about $29,100$ events and 474 imbalanced labels, demonstrating, for the first time, scalability and practical superiority over traditional causal discovery baselines. We also perform ablation on scoring criteria, frequency thresholds, and Transformer quality.

2 Preliminary & Related Work

A full description of the notations and definitions used throughout the paper can be found in Appendix A.

Event Sequence Modelling. Event sequences are typically represented as a series of time-stamped discrete events $S = \{(t_1, x_1), \ldots, (t_L, x_L)\}$ where $0 \leq t_1 < \ldots \leq t_L$ denotes the time of occurrence of event type $x_i \in \mathbb{X}$ drawn from a finite vocabulary $\mathbb{X}$. In multi-label settings, a binary label vector $\boldsymbol{y} \in \{0,1\}^{|\mathbb{Y}|}$ is attached to S and indicates the presence of multiple outcome labels chosen from $\mathbb{Y}$ occurring at the final time step t_L. Together, this results in a multi-labeled sequence $S_l = (S, (\boldsymbol{y}_L, t_L))$.

Event sequence modelling has been widely applied to predictive tasks. For instance, in the automotive domain, Diagnostic Trouble Codes (DTCs) [34] are logged asynchronously over time and used to infer failures or error patterns [28]. In healthcare, electronic health records encode temporal sequences of symptoms to perform predictive tasks [39, 20, 15]. A common modelling strategy [22, 29] separates such event types $\mathbb{X}$ from labels $\mathbb{Y}$.

Transformers [48, 38, 45] have emerged as the dominant architecture for sequence modelling. Recent work leveraged Transformers in high-dimensional event spaces for next-event and label prediction. Math et al. [28] proposed a dual Transformer architecture in which one model predicts the next event type (DTC) and the other predicts label occurrence (e.g., error patterns). Through this paper, we repurpose this dual architecture for causal discovery.

Multi-label Causal Discovery seeks to identify the Markov Boundary (**MB**) of each label—its minimal set of parents, children, and spouses—such that the label is conditionally independent of all other variables given its **MB** [46] (Def. 3).

While classical constraint-based algorithms have shown success on low-dimensional tabular data [41, 52], their application to event sequences with multi-label outputs remains challenging due to: (1) *dimensionality*—thousands of event types increase super-exponentially the number of graphs (2) *sparsity*—multi-hot encodings often underrepresent rare but important events (3) *distributional assumptions*—such as linearity or Gaussian noise, which rarely hold in real-world sequences [12].

Contemporary work points to decomposing classical causal discovery for high-dimensional datasets into sub-problems and graph aggregation. Laborda et al. [21] introduce a ring-based distributed algorithm for learning high-dimensional BN, Dong et al. [6] explores a distributed approach for large-scale causal structure learning and Mokhtarian et al. [30] for Markov Boundaries.

Bayesian Network [33] has served as a modelling technique for a variety of decision problems. It is defined as a triplet $< \boldsymbol{U}, \mathbb{G}, P >$ with P the joint distribution over a variable set $\boldsymbol{U}$ of a directed acyclic graph $\mathbb{G} = (\boldsymbol{U}, E)$ with E as the set of directed edges. This triplet must satisfy the Markov Condition: every random variable U_i is independent of its non-descendant variables given its parents $\text{Pa}(U_i)$ in $\mathbb{G}$. The directed edge $(U_i \rightarrow U_j)$ encodes a probabilistic dependence. Thus, the joint probability distribution can be factorized as:

$$P(U_1, \cdots, U_n) = \prod_{i=1}^{n} P(U_i | \text{Pa}(U_i))$$

The DAG encodes a set of conditional independencies $\mathcal{I}(\mathbb{G})$, where each element corresponds to a conditional independence relation $U_i \perp U_j | \boldsymbol{Z}$, meaning that U_i and U_j are conditionally independent

given the set of variables $\boldsymbol{Z}$. Formally, a DAG $\mathbb{G}_k$ is an I-map (or Independence map) of another $\mathbb{G}$ if the set of conditional independencies encoded by $\mathbb{G}_k$ is a subset of those encoded by $\mathbb{G}$:

$$\mathcal{I}(\mathbb{G}_k) \subseteq \mathcal{I}(\mathbb{G})$$

$\mathbb{G}_k$ is a minimal I-map of $\mathbb{G}$ if removing any edge from it introduces a conditional dependence that would violate an independence in $\mathbb{G}$, i.e $\mathcal{I}(\mathbb{G}_k \backslash \{e\} \not\subseteq I(\mathbb{G}) \forall e \in E$.

Bayesian Model Averaging. Fusing BN has several direct applications. Either to average multiple models from different experts to learn a global and average representation [16]. Or to perform causal discovery in distributed settings with federated learning algorithms [50, 13].

Formally, Given a set of Bayesian Networks $\{B_k\}_{k=1}^m$ with associated DAGs $\{\mathbb{G}_k = (V_k, E_k)\}_{k=1}^m, V_k \in \boldsymbol{U}$ sharing the same finite set of node $\boldsymbol{U}$, structural fusion aim to construct the DAG $\mathbb{G}^* = (V, E), V \in \boldsymbol{U}$. Multiple fusion methods exist and leverage either the probability distribution p by doing Bayesian Model Averaging [16] or focus on the structural learning (Fig. 8) of $\mathbb{G}^*$ [5, 32, 11, 35, 13], which is an NP-hard [32] problem.

We will focus on the second element in this paper. Hence, we are seeking the merged edges $E = \bigcup_{i=1}^m E_i^\sigma$. The consistent node ordering σ ensures acyclicity. The fused DAG $\mathbb{G}^*$ is the minimal I-map of the intersection of the conditional independencies across all DAGs $\mathbb{G}_k = (V_k, E_k)\}_{k=1}^m$.

Greedy Equivalence Search. (GES) [2] is one of the most theoretically sound methods to recover a Markov equivalence class (MEC, Def. 4) of a DAG. In large-sample settings, GES provides theoretical guarantees of recovering the true graph. Formally, GES searches for the MEC of the graph $\mathbb{G}^*$ from the observational dataset $\boldsymbol{D}$ with distribution p. This defines the optimization problem as:

$$\mathbb{G}^* = \arg_{\mathbb{G}} \max S(\mathbb{G}, \boldsymbol{D}) \tag{1}$$

Chickering [2] proved that under parametric assumption, a large number of samples and using the Bayesian Information Criterion (BIC) as criterion S, GES is guaranteed to recover a Markov Equivalence Class of $\mathbb{G}^*$.

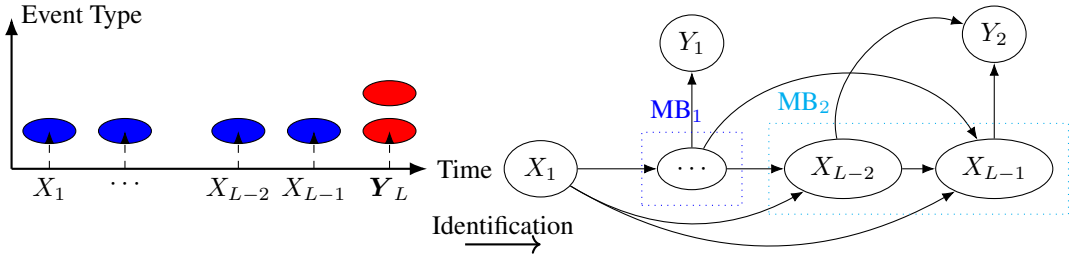
Frequency Based. Multiple heuristics have been developed to merge multiple BNs. One is edge frequency cutoff [43], other on estimating the proportion of false positive edges [8] and integer linear programming (ILP) [8] or a mix of both: [44]. As pointed out by [8], choosing a frequency cutoff τ is particularly challenging. Moreover, under class imbalance and long-tail problem in classification [53], the theoretical property of frequency approaches, such as the law of large numbers, might not hold.

We will now explain the two phases of CARGO, which are (1) One-shot causal discovery (2) Graph aggregation using adaptive thresholds. The Proofs of Lemmas and Theorems can be found in the Appendix B.

3 One-Shot Causal Discovery

Let S_l^k be a multi-labeled sequence drawn from a dataset $D = \{S_l^1, \cdots, S_l^m\} \subset \mathbb{S}$ and $\mathbb{G}_k$ the sequential BN (Fig. 1) with attached labels. The goal of multi-label causal discovery is to identify the

Figure 1: An example of a causal graph extracted from a single multi-label event sequence where MB_1 represents the Markov Boundary of Y_1 and MB_2 the Markov Boundary of Y_2.



(a) Multi-label event sequence S_l (b) BN representation of S_l

Markov Boundary of each label $Y_j \in \boldsymbol{Y}$ present in S_l^k. To be able to access conditional independence (Def. 2) between X_i and Y_j conditioned on the past events $\boldsymbol{Z} = (x_1, \cdots, x_{i-1}) = S_{<i}$, we model the event apparitions using a sequential BN (Fig. 1).

We want to assess how much additional information event X_i occurring at step i provides about label Y_j when we already know the past sequence of events $\boldsymbol{Z} = S_{<i}$. We essentially try to answer if:

$$P(Y_j|X_i, \boldsymbol{Z}) = P(Y_j|\boldsymbol{Z}) \Leftrightarrow D_{KL}(P(Y_j|X_i, \boldsymbol{Z})\|P(Y_j|\boldsymbol{Z})) = 0$$

where D_{KL} denotes the *Kullback-Leibler divergence* [3]. The distributional difference between the conditionals $P(Y_j|X_i, \boldsymbol{Z}), P(Y_j|\boldsymbol{Z})$ is akin to Information Gain I_G [37] conditioned on past events:

$$I_G(Y_j, x_i|z) \triangleq D_{KL}(P(Y_j|X_i = x_i, \boldsymbol{Z} = z))||P(Y_j|\boldsymbol{Z} = z)) \tag{2}$$

Which is equals to the difference between the conditional entropies [3] denoted as H:

$$I_G(Y_j, x_i|z) = H(Y_j|z) - H(Y_j|x_i, z) \tag{3}$$

More generally, we can access the conditional independence of event X_i and label Y_j using the conditional mutual information (CMI) [3] which is simply the expected value over z of the information gain $I_G(Y_j, X_i|z)$ such as:

$$I(Y_j, X_i|\boldsymbol{Z}) \triangleq H(Y_j|\boldsymbol{Z}) - H(Y_j|\boldsymbol{Z}, X_i) = \mathbb{E}_{p(z)}[I_G(Y_j, X_i|\boldsymbol{Z} = z)] \tag{4}$$

It can be interpreted as the expected value over all possible contexts $\boldsymbol{Z}$ of the deviation from independence of X_i, Y_j in this context. To approximate Eq. (4), a naive Monte Carlo [7] approximation is performed where we draw N random variations of the conditioning set $z^{(l)} = \{x_0^{(l)}, \ldots, x_{i-1}^{(l)}\}$, denoting the l-th sampled particle:

$$\hat{I}_N(Y_j, X_i \mid \boldsymbol{Z}) = \frac{1}{N} \sum_{l=1}^{N} I_G(Y_j, X_i \mid \boldsymbol{Z} = z^{(l)}) \tag{5}$$

This estimator is unbiased because the contexts $z^{(l)}$ are sampled directly from Tf_x using a proposal Q with the same support as $P(\boldsymbol{Z})$. Since $I_G(Y_j, X_i \mid \boldsymbol{Z} = z)$ is a difference between conditional entropies ((2)), it is thus bounded uniformly [3] by the log of supports such as:

$$0 < I_G(Y_j, X_i \mid \boldsymbol{Z} = z^{(l)}) = H(Y_j|z^{(l)}) - H(Y_j|X_i, z^{(l)})) \leq H(Y_j) \leq \log|\mathbb{Y}|$$

Thus the posterior variance of $f_i = I_G(Y_j, X_i \mid \boldsymbol{Z} = z^{(l)})$ satisfies $\sigma_{f_i}^2 \triangleq \mathbb{E}_{p(z)}[f_i^2(p(z)] - I^2(f_i) < +\infty$ [7] then the variance of $\hat{I}_N(f_i))$ is equal to $var(\hat{I}_N(f_i)) = \frac{\sigma_{f_i}^2}{N}$ and from the strong law of large numbers:

$$\hat{I}_N \xrightarrow[N \to +\infty]{\text{a.s.}} \mathbb{E}_{p(z)}[I_G(Y_j, X_i \mid \boldsymbol{Z} = z)] \triangleq I(f_i). \tag{6}$$

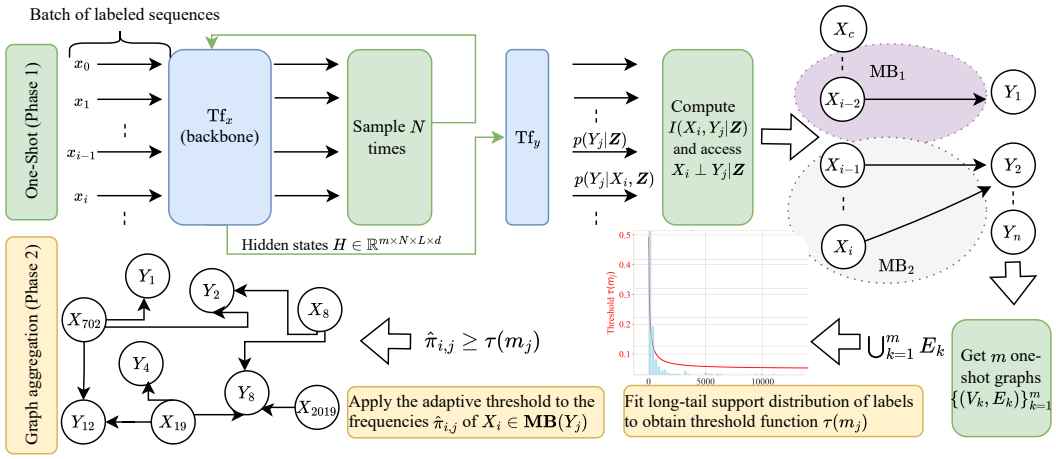
Density Estimation. We used two pretrained Transformers $(\text{Tf}_x, \text{Tf}_y)$ trained via maximum likelihood on a dataset of multi-labelled event sequences $D = \{S_l^1, \cdots, S_l^m\} \subset \mathbb{S}$. We assume that they perfectly model the true conditional distributions of events and labels (A4). While, due to the strict equivalences between CI-tests and conditional independence, it is difficult to provide a theoretical guarantee under imperfect models, we acknowledge that this assumption may be violated. We note that most causal discovery methods either impose strong parametric data assumptions or rely on a perfect CI-test. We provide an ablation study on the impact of the NADE's quality on the one-shot phase in Appendix C.1, including the number of parameters, context c, and Tf_y performance.

Formally, the two Transformers infer the probability of the next event and label conditioned on the past events using the hidden states $\boldsymbol{h}_{i-1}^x, \boldsymbol{h}_i^y \in \mathbb{R}^d$, from Tf_x, Tf_y respectively:

$$\text{Tf}_x(S_{<i}) = \textit{Softmax}(\boldsymbol{h}_{i-1}^x) = P_{\theta_x}(X_i|\boldsymbol{Z}) \tag{7}$$

$$\text{Tf}_y(S_{\leq i}) = \textit{Sigmoid}(\boldsymbol{h}_i^y) = P_{\theta_y}(Y|X_i, \boldsymbol{Z}) \tag{8}$$

Figure 2: The overview of CARGO. Phase 1 (One-shot) is on top, and Phase 2 (Adaptive Thresholding) is on the bottom. d denotes the hidden dimension, L the sequence length, m the number of samples and MB_1, MB_2 the Markov Boundary of Y_1, Y_2. All green and blue areas are parallelized.



Sequential One-shot Causal Discovery. The CMI using Eq.(4) is computable only with the posteriors $P(Y_j|\boldsymbol{Z}), P(Y_j|X_i, \boldsymbol{Z}))$. In practice a label-specific threshold $\theta_j \approx 0$ is applied to Eq. (4) to identify conditional dependence:

$$Y_j \not\perp X_i \mid \boldsymbol{Z} \quad \Leftrightarrow \quad I(Y_j, X_i \mid \boldsymbol{Z}) > \theta_j \approx 0. \tag{9}$$

Hence, the expectation in Eq.(4) is computed using a Monte-Carlo simulation, by sampling N similar context $\boldsymbol{Z}$ from Tf_x. Such that for each position in the sequence, we generate N plausible next tokens using a combination of top-k and nucleus sampling [17]. Ablation studies on the effect of the sampling method and thresholds are given in Appendix C.2, C.3.

Theorem 1 (Markov Boundary Identification in Event Sequences)**.** *If S_l^k a multi-labeled sequence drawn from a dataset $D = \{S_l^1, \cdots, S_l^m\} \subset \mathbb{S}$ where two Oracle Models Tf_x and Tf_y were trained on, then under causal sufficiency (A3), bounded lagged effects (A2) and temporal precedence (A1), the Markov Boundary of each label Y_j in the causal graph $\mathbb{G}$ can be identified using conditional mutual information for CI-testing.*

Theorem 1 enables us to sequentially recover the Markov Boundary of each label in a sequence. It provides a theoretical guarantee to recover the correct causes for each label Y_j.

Computation. A key advantage of our approach is its scalability. Unlike traditional methods whose complexity depends on the event and label cardinality $|\mathbb{X}|$ and $|\mathbb{Y}|$ [23], phase 1 is agnostic to both. Figure 2 shows all parallelized steps on GPUs. CMI estimations are independently performed for all positions $i \in [c, L]$, with the sampling pushed into the batch dimension and results averaged across labels. This transitions the time complexity from $\mathcal{O}(\text{BS} \times N \times L)$ to $\mathcal{O}(1)$ per batch, with L being the sequence length.

To ensure stable conditional entropy estimates and reliable predictions from Tf_y, the CMI is computed after observing c events (*context*). This design choice also enables out-of-the-box parallelization. By sampling N variations of the prefix sequence $S_{\leq c}$, the CMI is independently computed across positions $i \in [c, L]$. In our experiments, we set $c = 15, L = 192$. The implementation of Phase 1 in *PyTorch* [31] is provided in Appendix D.2.

4 Structural Fusion of Markov Boundaries

Let $\{\mathbb{G}_k = (V_k, E_k)\}_{k=1}^m, V_i \in \boldsymbol{U}$ be the set of DAGs generated by the Phase 1 from the dataset $\boldsymbol{D}$ containing m i.i.d sequences $\{S_l^k\}_{k=1}^m$ drawn from a joint distribution $p(x, y)$. Each graph $\mathbb{G}_k$ represents local Markov Boundaries identified within sequence S_l^k. Our objective is to fuse these

local graphs into a single, global consensus graph $\mathbb{G}^* = (\boldsymbol{U}, E)$ (see Fig. 8) with the events always as parents of labels Y_j such as:

$$\text{Pa}(Y_j) \subseteq \{X_1, \cdots, X_n\}$$

A naive fusion approach, such as taking the simple union of all edges $E = \bigcup_{k=1}^{m} E_k^\sigma$ works iff the Oracle models yield the perfect CI-tests in Phase 1 and thus the local graphs $\mathbb{G}^k$ are faithful to $p(x, y)$ ([32], Theorem 4.). Here, the ordering σ doesn't matter since we are dealing with Markov Boundaries. Moreover, G^* is naturally a DAG because we considered previously that outcome labels are solely explained by events, which simplifies acyclicity. We define the Bernoulli variable $Z_{i,j}^k$ for each potential edge $(X_i \to Y_j)$ within each sequence S_l^k:

$$Z_{i,j}^k = \begin{cases} 1 \text{ if the edge } X_i \to Y_j \text{ is present in } \mathbb{G}_k \\ 0 \text{ otherwise} \end{cases}$$

Under the Oracle Models assumption (A4), our one-shot discovery phase serves as a perfect conditional-independence tester. Consequently, the detection of an edge in a local graph $\mathbb{G}_k$ corresponds precisely to a true causal dependency in the global graph $\mathbb{G}^*$. The probability of this event $P(Z_{i,j}^k = 1)$ is therefore the true marginal probability of the edge's existence, which we denote as $\pi_{i,j}$.

The empirical frequency, $\hat{\pi}_{i,j}(m)$, of the edge $(X_i \to Y_j)$ after observing m sequences is the sample mean of these i.i.d Bernoulli variables $\hat{\pi}_{i,j}(m) = \frac{1}{m}\sum_{k=1}^{m} Z_{ij}^k$. By the Law of Large Numbers (LLN), as the number of i.i.d sequences m tends to infinity, the empirical frequency converges in probability to the true expected value of the random variable:

$$\hat{\pi}_{i,j}(m) \xrightarrow{p} \mathbb{E}[Z_{i,j}^{(k)}] = \pi_{i,j}$$

Thus, given a sufficiently large number of sequences, the empirical frequency $\hat{\pi}_{i,j}(m)$ serves as a consistent estimator for the true probability of the edge's existence in the global DAG $\mathbb{G}^*$.

4.1 Aggregation under imperfect CI-tests

The assumption of an Oracle CI-tester, while necessary for initial theoretical guarantees, is invariably violated in practice due to factors like model capacity, limited data, or class imbalance. The extracted one-shot graphs will most likely violate the independencies in $\mathbb{G}_k$ and thus $\mathbb{G}^*$.

Let us model the performance of our one-shot CI-test for any potential edge $X_i \to Y_j$ with the following error rates: (1) False Positive Rate (Type I Error): $\alpha = P(\text{detect} \mid \text{edge is spurious})$ (2) True Positive Rate (Sensitivity): $1 - \beta = P(\text{detect} \mid \text{edge is causal})$. We operate under the reasonable assumption that our one-shot classifier is significantly better than random, which implies that $1 - \beta \gg \alpha$. The expected value of our Bernoulli variable $Z_{i,j}^k$ is now:

$$\begin{aligned} \mathbb{E}[Z_{ij}^k] &= P(Z_{ij}^k = 1) \\ &= P(\text{detect} \mid \text{causal})P(\text{causal}) + P(\text{detect} \mid \text{spurious})P(\text{spurious}) \\ &= (1 - \beta)\pi_{ij} + \alpha(1 - \pi_{ij}) \end{aligned}$$

The empirical frequency now converges to this new expectation. For a true edge ($\pi_{i,j} = 1$), the empirical frequency converges to a high value: $\hat{\pi}_{ij}(m) \xrightarrow{p} 1 - \beta$ and for a spurious edge ($\pi_{i,j} = 0$) it converges to a low value: $\hat{\pi}_{ij}(m) \xrightarrow{p} \alpha$ This reveals the critical role of frequency aggregation as a mechanism for separating signal from noise.

4.2 Adaptive Fusion for Structural Discovery in Long-Tail Distributions

A primary challenge in real-world causal discovery is the long-tail distribution of outcome labels, where a few "head" labels possess abundant data while the vast majority of "tail" labels are data-sparse [53].

For rare labels, where empirical edge frequencies are high-variance estimators, a conservative high threshold is necessary to maintain precision against statistical noise. Conversely, for common labels where frequencies are reliable, a high threshold would be overly stringent, purging weaker but valid

Figure 3: Example of an error pattern (y_1) defined as a boolean rules of diagnosis trouble codes (x_i)

$$y_1 = x_1 \ \& \ x_5 \ \& \ x_8 \ \& \ x_{18} \ \& \ x_{12} \ \& \ x_3 \ \& \ !x_{10} \ \& \ !x_{20}$$

causal links. To resolve this, we introduce an adaptive thresholding strategy (Fig. 7) that tailors the edge inclusion criterion to the statistical power available for each label. We define a label-specific threshold τ_j, as a logistic decay function of its sample support m_j:

$$\tau_j(m_j) = (\tau_{\max} - \tau_{\min}) \cdot \frac{1}{1 + e^{k(\log m_j - \log m_0)}} + \tau_{\min} \tag{10}$$

This function smoothly interpolates between a user-defined maximum threshold, τ_{max} (prioritizing precision for the tail), and a minimum, τ_{min} (prioritizing recall for the head). Crucially, the function's behavior is calibrated by the distribution of the data. Such that decay midpoint m_0 is set to the median of all label supports, providing a robust anchor point against skew.

The decay rate, k is made inversely proportional to the log-inter-quartile range of supports such that $k = \frac{2 \log 3}{\log q_{75} - \log q_{25}}$. For rare labels with small m_j, the high variance of the frequency estimate necessitates a high threshold $\tau_j(m_j)$ that acts as a strong regularizer. For common labels with large m_j, the LLN guarantees the convergence of $\hat{\pi}_{i,j}$ to the true edge probability, justifying a lower threshold to capture a more complete causal structure. Hence, this strategy serves as a data-driven denoising mechanism and shares theoretical parallels with ensembling methods [1, 55], thereby enhancing the robustness and accuracy of the final fused graph.

5 Empirical Evaluation

Settings & Vehicle Dataset. We used a $g4dn.12xlarge$ instance from AWS Sagemaker to run comparisons. It contains 48 vCPUs and 4 NVIDIA T4 GPUs. We used a combination of F1-Score, Precision, and Recall with different averaging methods [54] to compare results. We evaluated our method on a real-world vehicular test set of $m = 300,000$ sequences, with $|\mathbb{Y}| = 474$ different error patterns and $|\mathbb{X}| = 29,100$ different DTCs forming sequences of $\approx 100 \pm 35$ events. We used Tf_x and Tf_y with 90m and 15m parameters [28]. The two NADEs didn't see the test set during training. Domain experts manually define error patterns as Boolean rules among DTCs (Fig. 3). We set the elements of this rule as the correct Markov Boundary for each label y_j in the tested sequences.

Multi-label Causal Discovery Comparisons. We benchmark CARGO against local structure learning (LSL) algorithms that estimate Markov Boundaries. This includes established approaches such as CMB [9], MB-by-MB [49], PCD-by-PCD [51], IAMB [47] from the ***PyCausalFS*** package [52], as well as the more recent, state-of-the-art MI-MCF [26]. 6 random folds of the test data were created and converted into a multi-one-hot data-frame where one row represents one sequence, and each column represents an event type or label $(\mathbb{X}, \mathbb{Y})$.

Ablation on Aggregation Criterions for Phase 2. We provide an Ablation of the different criteria used in the structural fusion of Markov Boundaries. Union stands for a simple union over all edges without any removal. Frequency or edge voting, counts how often is $X_i \in \text{MB}(Y_j)$. Then apply a static frequency threshold $\tau = [0.05, 0.25, 0.5, 0.8]$. MI uses the mutual information between events and labels as a criterion in a Background Equivalence Search [2]. Expected FPR (false positive ratio) [8] describes two beta distributions which are fitted using the distribution of the mutual information $I(X_i, Y_j)$ extracted from Phase 1. The lower tail is used for outlier detection. Different FPR are chosen $\beta = [0.01, 0.05, 0.15, 0.2]$. Detailed definitions can be found in Appendix C.4.

5.1 Results

Comparisons. We performed comparisons on Table 1 with $n = 50,000$ random sequences. We found that even under this reduced setup, LSL algorithms failed to compute the Markov Boundaries within 3 days (a 3-day timeout), far exceeding practical deployment limits. This behavior highlights the current infeasibility of multi-label causal discovery in high-dimensional event sequences. Current algorithms are cursed under high-dimensional data, since they rely on expensive CI testing that scales

quadratically with the number of nodes [12]. This positions CARGO as a more feasible approach for large-scale, multi-label causal discovery.

Table 1: Comparisons of **MB** retrieval with $m = 50,000$ samples averaged over $6-$folds with $|Y| = 474, |X| = 29,100$ nodes. Averaging is 'weighted'. The symbol '-' indicates that the algorithm didn't output the **MBs** within 3 days. Metrics are given in $\%$.

Algorithm	**Precision**$\uparrow$	**Recall**$\uparrow$	**F1**$\uparrow$	**Running Time (min)**$\downarrow$
IAMB	-	-	-	> 4320
CMB	-	-	-	> 4320
MB-by-MB	-	-	-	> 4320
PCDbyPCD	-	-	-	> 4320
MI-MCF	-	-	-	> 4320
CARGO	$\mathbf{60.6 \pm 1.5}$	$\mathbf{45.8 \pm 1.7}$	$\mathbf{45.8 \pm 1.2}$	$\mathbf{11.7}$

Criterions. Figure 4 illustrates the impact of aggregation choices during Phase 2. A naïve Union maximizes recall (84% for weighted) but suffers from poor precision. When optimizing a local scoring criterion based on the mutual information BES mi, it didn't significantly improve performance over a basic Union.

Moreover, instead of optimizing a score, fitting Beta distributions to detect outliers using their mutual information appears to perform better; hence, frequency beta outperforms alternatives, particularly in terms of lower FPR. Frequency approaches with a static threshold confirms the analysis in Section 4.1. When a large number of samples per class m_j is available, the frequency cut-off τ needs to be lower to not penalize classes with big support. Thus, we see that frequency with $\tau = [0.5, 0.8]$ have the lower weighted f1 score of all criterions. On the other hand, a small cut-off $\tau = [0.05, 0.25]$ enables a huge improvement in the weighted average ($+40\%$ precision), but decrease its macro average metrics (-20% in precision).

Finally, our adaptive thresholding criterion leverages a small threshold for big supports and a big threshold for small supports, which takes advantage of the long-tail distribution. As a result, it is first on both averaging, with respectively 44.88% and 40.9% for weighted and macro f1 score, and 62.8% and 66.1% for weighted and macro precision.

6 Conclusion

We introduced CARGO, a novel framework for multi-label causal discovery in high-dimensional event sequences. By combining one-shot causal discovery with adaptive frequency-aware aggregation, CARGO successfully recovers interpretable causal structures from noisy observational data—achieving results in minutes where classical methods fail to scale.

CARGO could scale further by leveraging general-purpose foundation models for sequences (e.g., time-series transformers pretrained across domains). Such models could extend the applicability beyond automotive diagnostics to healthcare, cybersecurity, and other structured domains.

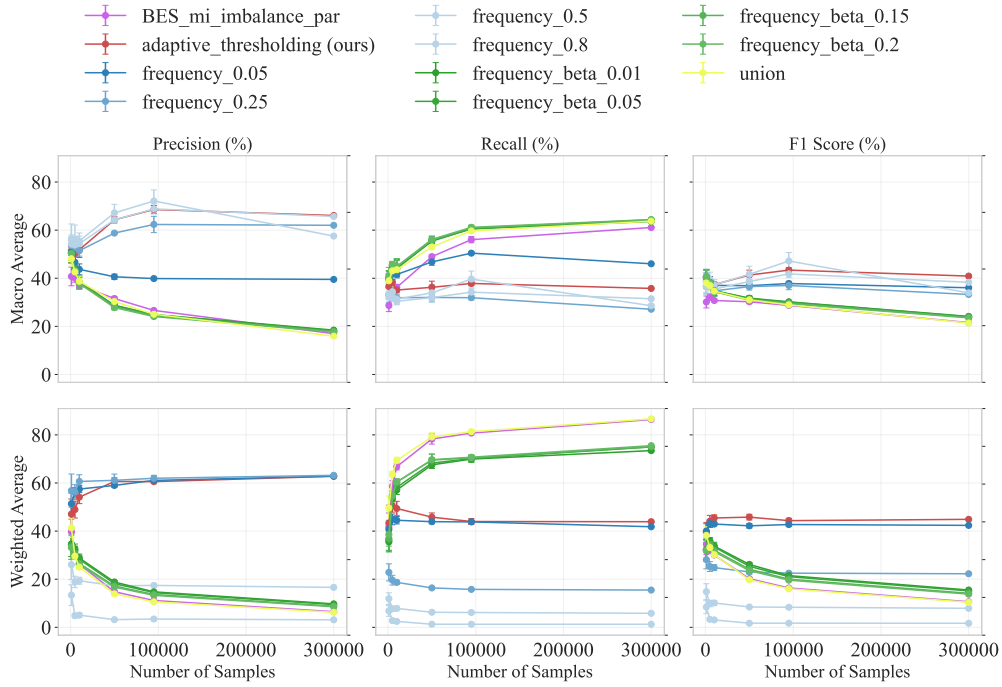
Under the temporal assumptions, large samples, faithfulness, and perfect CI-tests, CARGO recovers the true set of Markov Boundaries. However, we underlined the practical limitations, in particular, long-tail distributions, a common trait in high-dimensional labeled data. Future work should extend this framework to support event-to-event causality and, if possible, relax assumptions such as bounded lagged effects or temporal precedence.

Ultimately, CARGO demonstrates how structured probabilistic methods can bridge the gap between causal discovery theory and scalable, practical deployment in complex industrial systems.

References

[1] L. Breiman. Random forests. Mach. Learn., 45(1):5–32, Oct. 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Figure 4: Comparison of different criteria for the structural fusion (Phase 2) in function of the number of samples m. With $|Y| = 474, |X| = 29,100$ nodes.



[2] D. M. Chickering. Optimal structure identification with greedy search. J. Mach. Learn. Res., 3(null):507–554, Mar. 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717. URL https://doi.org/10.1162/153244303321897717.

[3] T. Cover. Elements of Information Theory. Wiley series in telecommunications and signal processing. Wiley-India, 1999. ISBN 9788126508143. URL https://books.google.de/books?id=3yGJrqyanyYC.

[4] T. Cover and J. Thomas. Elements of Information Theory. Wiley-Interscience, 2 edition, 2006. URL https://www.amazon.com/Elements-Information-Theory-Thomas-Cover/dp/0471241954.

[5] J. del Sagrado and S. Moral. Qualitative Aggregation of Bayesian Networks, pages 91–108. Springer Vienna, Vienna, 2001. ISBN 978-3-7091-2580-9. doi: 10.1007/978-3-7091-2580-9_5. URL https://doi.org/10.1007/978-3-7091-2580-9_5.

[6] S. Dong, M. Sebag, K. Uemura, A. Fujii, S. Chang, Y. Koyanagi, and K. Maruhashi. DCILP: A distributed approach for large-scale causal structure learning. In T. Walsh, J. Shah, and Z. Kolter, editors, AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, pages 16345–16353. AAAI Press, 2025. doi: 10.1609/AAAI.V39I15.33795. URL https://doi.org/10.1609/aaai.v39i15.33795.

[7] A. Doucet, N. de Freitas, and N. Gordon. An Introduction to Sequential Monte Carlo Methods, pages 3–14. Springer New York, New York, NY, 2001. ISBN 978-1-4757-3437-9. doi: 10.1007/978-1-4757-3437-9_1. URL https://doi.org/10.1007/978-1-4757-3437-9_1.

[8] H. Fröhlich and G. W. Klau. Reconstructing Consensus Bayesian Network Structures with Application to Learning Molecular Interaction Networks. In T. Beißbarth, M. Kollmar, A. Leha, B. Morgenstern, A.-K. Schultz, S. Waack, and E. Wingender, editors, German Conference on Bioinformatics 2013, volume 34 of Open Access Series in Informatics (OASIcs), pages 46–55, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN

978-3-939897-59-0. doi: 10.4230/OASIcs.GCB.2013.46. URL https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.GCB.2013.46.

[9] T. Gao and Q. Ji. Local causal discovery of direct causes and effects. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/fcdf25d6e191893e705819b177cddea0-Paper.pdf.

[10] S. Garrido, S. Borysov, J. Rich, and F. Pereira. Estimating causal effects with the neural autoregressive density estimator. Journal of Causal Inference, 9(1):211–228, 2021. doi: doi: 10.1515/jci-2020-0007. URL https://doi.org/10.1515/jci-2020-0007.

[11] J. Go and T. Isaac. Robust expected information gain for optimal bayesian experimental design using ambiguity sets. In The 38th Conference on Uncertainty in Artificial Intelligence, 2022. URL https://openreview.net/forum?id=HU9IxO8oqlc.

[12] C. Gong, C. Zhang, D. Yao, J. Bi, W. Li, and Y. Xu. Causal discovery from temporal data: An overview and new perspectives. ACM Comput. Surv., 57(4), Dec. 2024. ISSN 0360-0300. doi: 10.1145/3705297. URL https://doi.org/10.1145/3705297.

[13] X. Guo, K. Yu, L. Liu, and J. Li. Fedcsl: A scalable and accurate approach to federated causal structure learning. Proceedings of the AAAI Conference on Artificial Intelligence, 38 (11):12235–12243, Mar. 2024. doi: 10.1609/aaai.v38i11.29113. URL https://ojs.aaai.org/index.php/AAAI/article/view/29113.

[14] U. Hasan, E. Hossain, and M. O. Gani. A survey on causal discovery methods for i.i.d. and time series data. Transactions on Machine Learning Research, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=YdMrdhGx9y. Survey Certification.

[15] W. He, X. Mao, C. Ma, Y. Huang, J. M. Hernàndez-Lobato, and T. Chen. Bsoda: A bipartite scalable framework for online disease diagnosis. In Proceedings of the ACM Web Conference 2022, WWW '22, page 2511–2521, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512123. URL https://doi.org/10.1145/3485447.3512123.

[16] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: a tutorial. Statistical Science, 14(4):382–417, 1999.

[17] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

[18] D. Janzing, D. Balduzzi, M. Grosse-Wentrup, and B. Schölkopf. Quantifying causal influences. The Annals of statistics, 41(5):2324–2358, 2013. ISSN 0090-5364.

[19] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. Phys. Rev. E, 69:066138, Jun 2004. doi: 10.1103/PhysRevE.69.066138. URL https://link.aps.org/doi/10.1103/PhysRevE.69.066138.

[20] A. Labach, A. Pokhrel, X. S. Huang, S. Zuberi, S. E. Yi, M. Volkovs, T. Poutanen, and R. G. Krishnan. Duett: Dual event time transformer for electronic health records. In K. Deshpande, M. Fiterau, S. Joshi, Z. Lipton, R. Ranganath, I. Urteaga, and S. Yeung, editors, Proceedings of the 8th Machine Learning for Healthcare Conference, volume 219 of Proceedings of Machine Learning Research, pages 403–422. PMLR, 11–12 Aug 2023. URL https://proceedings.mlr.press/v219/labach23a.html.

[21] J. D. Laborda, P. Torrijos, J. M. Puerta, and J. A. Gámez. A ring-based distributed algorithm for learning high-dimensional bayesian networks. In Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 17th European Conference, ECSQARU 2023, Arras, France, September 19–22, 2023, Proceedings, page 123–135, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-45607-7. doi: 10.1007/978-3-031-45608-4_10. URL https://doi.org/10.1007/978-3-031-45608-4_10.

[22] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

[23] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. CoRR, abs/1601.07996, 2016. URL http://arxiv.org/abs/1601.07996.

[24] M. Liu, C.-W. Lee, X. Sun, X. Yu, Y. QIAO, and Y. Wang. Learning causal alignment for reliable disease diagnosis. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=ozZG5FXuTV.

[25] Q. Luo, L. Zhang, Z. Xing, H. Xia, and Z.-X. Chen. Causal discovery of flight service process based on event sequence. Journal of Advanced Transportation, 2021(1):2869521, 2021. doi: https://doi.org/10.1155/2021/2869521. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/2869521.

[26] L. Ma, L. Hu, Y. Li, W. Ding, and W. Gao. Mi-mcf: A mutual information-based multilabel causal feature selection. IEEE Transactions on Neural Networks and Learning Systems, pages 1–15, 2025. doi: 10.1109/TNNLS.2025.3556128.

[27] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann. Flowtransformer: A transformer framework for flow-based network intrusion detection systems. Expert Systems with Applications, 241:122564, 2024. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.122564. URL https://www.sciencedirect.com/science/article/pii/S095741742303066X.

[28] H. Math, R. Lienhart, and R. Schön. Harnessing event sensory data for error pattern prediction in vehicles: A language model approach. Proceedings of the AAAI Conference on Artificial Intelligence, 39(18):19423–19431, Apr. 2025. doi: 10.1609/aaai.v39i18.34138. URL https://ojs.aaai.org/index.php/AAAI/article/view/34138.

[29] A. McCallum, D. Freitag, and F. C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, page 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[30] E. Mokhtarian, S. Akbari, A. Ghassami, and N. Kiyavash. A recursive markov boundary-based approach to causal structure learning. In T. D. Le, J. Li, G. Cooper, S. Triantafyllou, E. Bareinboim, H. Liu, and N. Kiyavash, editors, Proceedings of The KDD'21 Workshop on Causal Discovery, volume 150 of Proceedings of Machine Learning Research, pages 26–54. PMLR, 15 Aug 2021. URL https://proceedings.mlr.press/v150/mokhtarian21a.html.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: an imperative style, high-performance deep learning library. Curran Associates Inc., Red Hook, NY, USA, 2019.

[32] J. M. Peña. Finding consensus bayesian network structures. J. Artif. Int. Res., 42(1):661–687, Sept. 2011. ISSN 1076-9757.

[33] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 1558604790.

[34] P. Pirasteh, S. Nowaczyk, S. Pashami, M. Löwenadler, K. Thunberg, H. Ydreskog, and P. Berck. Interactive feature extraction for diagnostic trouble codes in predictive maintenance: A case study from automotive domain. In Proceedings of the Workshop on Interactive Data Mining, WIDM'19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362962. doi: 10.1145/3304079.3310288. URL https://doi.org/10.1145/3304079.3310288.

[35] J. M. Puerta, J. A. Aledo, J. A. Gámez, and J. D. Laborda. Efficient and accurate structural fusion of bayesian networks. Information Fusion, 66:155–169, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.09.003. URL https://www.sciencedirect.com/science/article/pii/S156625352030364X.

[36] J. Qiao, R. Cai, S. Wu, Y. Xiang, K. Zhang, and Z. Hao. Structural hawkes processes for learning causal structure from discrete-time event sequences. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/633. URL https://doi.org/10.24963/ijcai.2023/633.

[37] J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81–106, 1986.

[38] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[39] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi. Med-bert: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction. NPJ Digit Med. 2021 May 20;4(1):86, abs/2005.12833, 2020. doi: 10.1038/s41746-021-00455-y.

[40] R. Y. Rohekar, Y. Gurwicz, and S. Nisimov. Causal interpretation of self-attention in pre-trained transformers. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=DS4rKySlYC.

[41] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. Social Science Computer Review, 9(1):62–72, 1991. doi: 10.1177/089443939100900106. URL https://doi.org/10.1177/089443939100900106.

[42] P. Spirtes, C. Glymour, and R. Scheines. Causation, prediction, and search, 2nd edition. In Causation, Prediction, and Search (Second Edition), 2001. URL https://api.semanticscholar.org/CorpusID:124969922.

[43] E. Steele and A. Tucker. Consensus and meta-analysis regulatory networks for combining multiple microarray gene expression datasets. Journal of Biomedical Informatics, 41(6):914–926, December 2008. ISSN 1532-0464. doi: 10.1016/j.jbi.2008.01.011. URL https://doi.org/10.1016/j.jbi.2008.01.011.

[44] P. Torrijos, J. M. Puerta, J. A. Gámez, and J. A. Aledo. Informed greedy algorithm for scalable bayesian network fusion via minimum cut analysis, 2025. URL https://arxiv.org/abs/2504.00467.

[45] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[46] I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In C. M. Bishop and B. J. Frey, editors, Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, volume R4 of Proceedings of Machine Learning Research, pages 300–307. PMLR, 03–06 Jan 2003. URL https://proceedings.mlr.press/r4/tsamardinos03a.html. Reissued by PMLR on 01 April 2021.

[47] I. Tsamardinos, C. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. pages 376–381, 01 2003.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[49] C. Wang, Y. Zhou, Q. Zhao, and Z. Geng. Discovering and orienting the edges connected to a target variable in a dag via a sequential local learning approach. Computational Statistics and Data Analysis, 77:252–266, 2014. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2014.03.003. URL https://www.sciencedirect.com/science/article/pii/S0167947314000802.

[50] Z. Wang, P. Ma, and S. Wang. Towards practical federated causal structure learning. In Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part II, page 351–367, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-43414-3. doi: 10.1007/978-3-031-43415-0_21. URL https://doi.org/10.1007/978-3-031-43415-0_21.

[51] J. Yin, Y. Zhou, C. Wang, P. He, C. Zheng, and Z. Geng. Partial orientation and local structural learning of causal networks for prediction. In I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A. Statnikov, editors, Proceedings of the Workshop on the Causation and Prediction Challenge at WCCI 2008, volume 3 of Proceedings of Machine Learning Research, pages 93–105, Hong Kong, 03–04 Jun 2008. PMLR. URL http://proceedings.mlr.press/v3/yin08a.html.

[52] K. Yu, X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, and X. Wu. Causality-based feature selection: Methods and evaluations. ACM Comput. Surv., 53(5), Sept. 2020. ISSN 0360-0300. doi: 10.1145/3409382. URL https://doi.org/10.1145/3409382.

[53] C. Zhang, G. Almpanidis, G. Fan, B. Deng, Y. Zhang, J. Liu, A. Kamel, P. Soda, and J. Gama. A systematic review on long-tailed learning. IEEE Transactions on Neural Networks and Learning Systems, PP:1–21, 02 2025. doi: 10.1109/TNNLS.2025.3539314.

[54] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. Knowledge and Data Engineering, IEEE Transactions on, 26:1819–1837, 08 2014. doi: 10.1109/TKDE.2013.39.

[55] Q.-C. Zheng, S.-H. Lyu, S.-Q. Zhang, Y. Jiang, and Z.-H. Zhou. On the consistency rate of decision tree learning algorithms. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, volume 206 of Proceedings of Machine Learning Research, pages 7824–7848. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/zheng23b.html.

A Notations and Definitions

A.1 Notations

We use capital letters (e.g., X) to denote random variables, lower-case letters (e.g., x) for their realisations, and bold capital letters (e.g., $\boldsymbol{X}$) for sets of variables. Let $\boldsymbol{U}$ denote the set of all (discrete) random variables. We define the event set $\boldsymbol{X} = \{X_1, \ldots, X_n\} \subset \boldsymbol{U}$, and the label set $\boldsymbol{Y} = \{Y_1, \ldots, Y_n\} \subset \boldsymbol{U}$. When explicitly said, event $X_i^{(t_i)}$ represent the occurrence of X_i at the sequence step i and time t_i. Similarly for $Y_{i+1}^{(t_{i+1})}$.

A.2 Definitions

Definition 1 (Faithfulness)**.** *Spirtes et al. [42]. Given a BN $< \boldsymbol{U}, \mathbb{G}, P >$, $\mathbb{G}$ is faithful to P if and only if every conditional independence present in P is entailed by $\mathbb{G}$ and the Markov condition holds. P is faithful if and only if there exist a DAG $\mathbb{G}$ such that $\mathbb{G}$ is faithful to P.*

Definition 2 (Conditional Independence)**.** *Variables X and Y are said to be conditionally independent given a variable set $\boldsymbol{Z}$, if $P(X, Y|\boldsymbol{Z}) = P(X|\boldsymbol{Z})P(Y|\boldsymbol{Z})$, denoted as $X \perp Y|\boldsymbol{Z}$. Inversely, $X \not\perp Y|\boldsymbol{Z}$ denotes the conditional dependence. Using the conditional mutual information [3] to measure the independence relationship, this implies that $\boldsymbol{I}(X, Y|\boldsymbol{Z}) = 0 \Leftrightarrow X \perp Y|\boldsymbol{Z}$.*

Definition 3 (Markov Boundary)**.** *Tsamardinos and Aliferis [46]. In a faithful BN $< \boldsymbol{U}, \mathbb{G}, P >$, for a set of variables $\boldsymbol{Z} \subset \boldsymbol{U}$ and label $Y \in \boldsymbol{U}$, if all other variables $X \in \{\boldsymbol{X} - \boldsymbol{Z}\}$ are independent of Y conditioned on $\boldsymbol{Z}$, and any proper subset of $\boldsymbol{Z}$ do not satisfy the condition, then $\boldsymbol{Z}$ is the Markov Boundary of Y: $\boldsymbol{MB}(Y)$.*

Definition 4. *(Markov Equivalence Class). Two distinct graphs $\mathbb{G}, \mathbb{G}'$ are said to belong to the same Markov Equivalence Class (MEC) if they have the same set of conditional independencies, i.e $I(\mathbb{G}) = I(\mathbb{G}')$.*

Definition 5. *(Decomposable Criterion). We say that a scoring criterion $S(\mathbb{G}, D)$ is decomposable if it can be written as a sum of measures, each of which is a function only of one node and its parents. In other words, a decomposable scoring criterion $\mathbb{S}$ applied to a DAG $\mathbb{G}$ can always be expressed as:*

$$S(\mathbb{G}, \boldsymbol{D}) = \sum_{i}^{n} s(X_i, \boldsymbol{Pa}_i^{\mathbb{G}}) \tag{11}$$

Definition 6 (Score equivalent)**.** *Chickering [2]. A score S is score equivalent if it assigns the same score to all the graphs in the same MEC.*

Definition 7 (Local Consistency)**.** *Chickering [2] Let $\boldsymbol{D}$ contain m iid samples from some distribution $p(.)$. Let $\mathbb{G}$ be any possible DAG and $\mathbb{G}'$ a different DAG obtained by adding the edge $i \rightarrow j$ to $\mathbb{G}$. A score S is locally consistent if both hold:*

- *If $X_i \not\perp_p X_j|Pa_j^{\mathbb{G}}$, then $S(\mathbb{G}', \boldsymbol{D}) > S(\mathbb{G}, \boldsymbol{D})$*
- *If $X_i \perp_p X_j|Pa_j^{\mathbb{G}}$, then $S(\mathbb{G}', \boldsymbol{D}) < S(\mathbb{G}, \boldsymbol{D})$*

A.3 Assumptions

Assumption 1 (Temporal Precedence)**.** *Given a perfectly recorded sequence of events $((x_1, t_1), \cdots, (x_L, t_L))$ with labels $(\boldsymbol{y}_L, t_L)$ and monotonically increasing time of occurrence $0 \leq t_1 \leq \cdots \leq t_L$, an event x_i is allowed to influence any subsequent event x_j such that $t_i \leq t_j$ and $i < j$. Formally, the graph $\mathbb{G} = (\boldsymbol{U}, \boldsymbol{E})$, $(x_i, x_j) \in \boldsymbol{E} \implies t_i \leq t_j$ and step $i < j$*

It allows us to remove ambiguity in causal directionality and is widely used in time-series and sequential data [12].

Assumption 2 (Bounded Lagged Effects)**.** *Once we observed events up to timestamp t_i and step i as $\boldsymbol{Z}_{\leq t_i} = ((x_1, t_1), \cdots, (x_i, t_i))$, any future lagged copy of event $X_i^{(t_i+\tau)}$ is independent of Y_j conditioned on $\boldsymbol{Z}_{\leq t_i}$:*

$$Y_j \perp X_i^{(t_i+\tau)}|\boldsymbol{Z}_{\leq t_i}$$

Where $\tau = t_{i+1} - t_i$ is a finite bound on the allowed time delay for causal influence.

In other words, we allow the causal influence of event X_i on Y_j until the next event X_{i+1} is observed. We note that for data with strong lagged effects (e.g., financial transactions), this might not hold well, but for log-based and error code-based data, this is usually correct.

Assumption 3 (Causal Sufficiency for Labels)**.** *All relevant variables are observed, and there are no hidden confounders affecting the labels.*

Assumption 4 (Oracle Models)**.** *We assume that two autoregressive Transformer models, Tf_x and Tf_y, are trained via maximum likelihood on a dataset of multi-labeled event sequences $D = \{S_l^1, \cdots, S_l^m\} \subset \mathbb{S}$, and can perfectly approximate the true conditional distributions of events and labels:*

$$P(X_i|Pa(X_i)) = P_{\theta_x}(X_i|Pa(X_i)) = Tf_x(S_{<i}),\ P(Y_j|Pa(Y_j)) = P_{\theta_y}(Y_j|Pa(Y_j)) = Tf_y(S_{\leq i}) \tag{12}$$

A.4 Lemmas

Lemma 1 (Identifiability of $\mathbb{G}$)**.** *Assuming the faithfulness condition holds for the true causal graph $\mathbb{G}$. Let Tf_x and Tf_y be oracle models that model the true conditional distributions of events and labels, respectively. The joint distribution P_{θ_x,θ_y} can then be constructed, and any conditional independence detected from the distributions estimated by Tf_x and Tf_y corresponds to a conditional independence in $\mathbb{G}$:*

$$X_i \perp_{\theta_x,\theta_y} Y_j \mid \boldsymbol{Z} \quad \Longrightarrow \quad X_i \perp_{\mathbb{G}} Y_j \mid \boldsymbol{Z}.$$

Where $\perp_{\theta_x,\theta_y}$ denotes the independence entailed by the joint probability P_{θ_x,θ_y}.

Lemma 2 (Markov Boundary Equivalence)**.** *In a multi-label event sequence S_l and under the temporal precedence assumption A1, the Markov Boundary of each label Y_j is only its parents such that $\forall X \in \{\boldsymbol{U} - Pa(Y_j)\}, X \perp Y_j|Pa(Y_j) \Leftrightarrow MB(Y_j) = Pa(Y_j)$.*

B Proofs

We provide proofs for the results described in Section 3

B.1 Proof of Lemma 1

Proof. We assume that the data is generated by the associated causal graph $\mathbb{G}$ following the sequential BN from a multi-labelled sequence S. And that the faithfulness assumption holds [33], meaning that all conditional independencies in the observational data are implied by the true causal graph $\mathbb{G}$.

Given that the Oracle models Tf_x and Tf_y are trained to perfectly approximate the true conditional distributions, for any variable U_i in the graph, we have:

$$P(U_i|\text{Pa}(U_i)) = \begin{cases} P(Y_j|\text{Pa}(Y_j)) = P_{\theta_y}(Y_j|\text{Pa}(Y_j)), & \text{if } U_i \in \boldsymbol{Y} \\ P(X_i|\text{Pa}(X_i)) = P_{\theta_x}(X_i|\text{Pa}(X_i)), & \text{otherwise.} \end{cases}$$

The joint distribution P_{θ_x,θ_y} can then be constructed using the chain rule $P_{\theta_x,\theta_y}(X_1, \cdots, X_i, Y_1, \cdots, Y_c) = \prod_{k=0}^{i} P(X_k|Pa(X_k)) \prod_l^c P(Y_l|\text{Pa}(Y_l))$. By the faithfulness assumption [33], if the conditional independencies hold in the data, they must also hold in the causal graph $\mathbb{G}$:

$$X_i \perp Y_j|\boldsymbol{Z} \implies X_i \perp_{\mathbb{G}} Y_j|\boldsymbol{Z}$$

Since we can approximate the true conditional distributions, it follows that:

$$X_i \perp_{\theta_x,\theta_y} Y_j|\boldsymbol{Z} \implies X_i \perp Y_j|\boldsymbol{Z} \implies X_i \perp_{\mathbb{G}} Y_j|\boldsymbol{Z}$$

Where $\perp_{\theta_x,\theta_y}$ denotes the independence entailed by the joint probability P_{θ_x,θ_y}. Thus, the graph $\mathbb{G}$ can be identified from the observational data. □

B.2 Proof of Lemma 2

Proof. Let $< \boldsymbol{U}, \mathbb{G}, P >$ be the sequential BN composed of the events from the multi-labeled sequence $S_l = (\{(t_1, x_1, \cdots, (t_L, x_L)\}_{i=1}^{L}, (\boldsymbol{y}_L, t_L))$. Following the temporal precedence assumption

A1, the labels $\boldsymbol{y_L}$ can only be caused by past events $(x_1, \cdots, x_L)$; moreover, by definition, labels do not cause any other labels. Thus, Y_j has no descendants, so no children and spouses. Therefore, together with the Markov Assumption, we know that $\forall X \in \{\boldsymbol{U} - Pa(Y_j)\} : Y_j \perp X | Pa(Y_j)$. Which is the definition of the MB (Def. 3). Thus, $\mathbf{MB}(Y_j) = Pa(Y_j)$.

□

B.3 Proof of Theorem 1.

Proof. By recurrence over the sequence length L of the multi-label sequence S_l^k, we want to show that under temporal precedence A1, bounded lagged effects A2, causal sufficiency A3, Oracle Models A4 the Markov Boundary of label Y_j can be identified in the causal graph $\mathbb{G}$.

Let's define $\mathcal{M}_j^L$ as the estimated Markov Boundary of Y_j after observing L events.

Base Case: L = 1: Consider the BN for step $L = 1$ following the Markov assumption [33] with two nodes X_1, Y_j. Using Tf_x, Tf_y as Oracle Models A4, we can express the conditional probabilities for any node U:

$$P(U|\text{Pa}(U)) = \begin{cases} P(X_1) = P_{\theta_x}(X_1|[CLS]) \text{ if } U \in \boldsymbol{X} \\ P(Y_j|X_1) = P_{\theta_y}(Y_j|X_1) \text{ otherwise} \end{cases} \tag{13}$$

Assuming that P is faithful (A1) to $\mathbb{G}$, no hidden confounders bias the estimate (A3) and temporal precedence (A1), we can estimate the CMI 4 such that iif $I(X_1, Y_j)|\emptyset) > 0 \Leftrightarrow Y_j \not\perp\!\!\!\perp_{\theta_x,\theta_y} X_1 \implies Y_j \not\perp\!\!\!\perp_{\mathbb{G}} X_1$ (Lemma 1).

Since we assume temporal precedence A1, we can orient the edge such that X_1 must be a parent of Y_j in $\mathbb{G}$. Using Lemma 2, we know that $Par(Y_j) = \mathbf{MB}(Y_j) \implies X_1 \in \mathbf{MB}(Y_j)$, thus we must include X_1 in M_j^1, otherwise not.

Heredity: For $L = i$, we obtained M_j^i with the sequential BN up to step $L = i$. Now for $L = i + 1$, the sequential BN has $i + 2$ nodes denoted as $\boldsymbol{U'} = (X_1, \cdots, X_i, X_{i+1}, Y_j)$. Using the Oracle Models A4 and following the Markov assumption [33], we can estimates the following conditional probabilities for any nodes $U \in \boldsymbol{U'}$:

$$P(U|\text{Pa}(U)) = \begin{cases} P(Y_j|\text{Pa}(Y_j)) \approx P_{\theta_y}(Y_j|\text{Pa}(Y_j)), & \text{if } U \in \boldsymbol{Y} \\ P(X|\text{Pa}(X)) \approx P_{\theta_x}(X|\text{Pa}(X)), & \text{otherwise.} \end{cases} \tag{14}$$

By bounded lagged effects (A2) we know that the causal influence of past $X_{\leq i}$ on Y_j has expired. Moreover, no hidden confounders (A3) bias the independence testing. Finally, using Eq. (4) we can estimate the CMI such that iif $I(Y_j, X_{i+1}|\boldsymbol{Z}) > 0 \Leftrightarrow Y_j \not\perp\!\!\!\perp_{\theta_x,\theta_y} X_{i+1}|\boldsymbol{Z} \implies Y_j \not\perp\!\!\!\perp_{\mathbb{G}} X_{i+1}|\boldsymbol{Z}$ (Lemma 1).

Since we assume temporal precedence A1, we can orient the edge so that X_{i+1} must be a parent of Y_j in $\mathbb{G}$. Using Lemma 2, we know that $Par(Y_j) = \mathbf{MB}(Y_j) \implies X_{i+1} \in \mathbf{MB}(Y_j)$. Thus $X_{i+1} \in M_j^{i+1}$ which represent the $\mathbf{MB}(Y_j)$ for step $i + 1$.

Finally, $\mathcal{M}_j^{i+1}$ still recovers the Markov Boundary of Y_j such that

$$\forall U \in \{\boldsymbol{U'} - \mathcal{M}_j^{i+1}\}, Y_j \perp U | \mathcal{M}_j^{i+1}$$

□

C Ablations

C.1 NADEs Quality.

We did several ablations on the quality of the NADEs and their impact on the one-shot causal discovery phase. In particular, Table 2 presents multiple Tf_x, Tf_y with respectively 90 and 15 million parameters or 34 and 4 million parameters. We also varied the context window (conditioning set $\boldsymbol{Z}$), trained on different amounts of data (tokens), and reported the classification results on the test set of Tf_y alone. We didn't output the running time since it was approximately the same for all NADEs: 1.27 minutes of 50,000 samples and 0.14 for 5000.

We observe that scaling up the NADEs model size and the trained data show the most significant improvements. Afterward, it is via the context c, which, after $c = 15$, shows a decline in performance across a larger number of samples. We then choose the backbone with 1.5B Tokens, 105m parameters, and a context $c = 15$ for our experiments.

Table 2: Ablations of the performance of Phase 1 (One-shot **MB** retrieval) in function of different NADEs with $m = 50{,}000$ and $m = 500$ samples averaged over 6-folds. Classification metrics use weighted averaging. Metrics are given in $\%$.

Tokens	**Parameters**	**Context**	**Precision (↑)**	**Recall (↑)**	**F1 Score (↑)**	**Tfy F1 (↑)**
			For $n = 50{,}000$ *samples*			
1.5B	105m	$c = 4$	47.95 ± 1.05	30.65 ± 0.51	37.39 ± 0.67	88.6
1.5B	105m	$c = 12$	54.62 ± 1.03	29.88 ± 0.73	38.63 ± 0.85	90.43
1.5B	105m	$c = 15$	$\mathbf{55.26 \pm 1.42}$	$\mathbf{31.37 \pm 0.82}$	$\mathbf{40.02 \pm 1.03}$	90.57
1.5B	105m	$c = 20$	49.52 ± 1.59	$\mathbf{31.76 \pm 0.85}$	36.54 ± 1.10	91.19
1.5B	105m	$c = 30$	36.65 ± 1.18	22.75 ± 0.78	26.57 ± 0.91	**92.64**
300m	47m	$c = 20$	39.49 ± 1.77	26.30 ± 0.89	29.01 ± 1.10	83.6
			For $n = 500$ *samples*			
1.5B	105m	$c = 12$	54.84 ± 4.55	$\mathbf{31.45 \pm 2.23}$	$\mathbf{39.95 \pm 2.83}$	90.43
1.5B	105m	$c = 15$	55.04 ± 3.36	29.90 ± 1.78	38.74 ± 2.24	90.57
1.5B	105m	$c = 20$	48.84 ± 4.01	$\mathbf{31.65 \pm 2.37}$	36.19 ± 2.65	**91.19**
300m	47m	$c = 20$	38.23 ± 2.91	25.31 ± 2.39	27.92 ± 2.25	83.6

C.2 Sampling Number

We tested different values of N for the sampling method across averaging methods (micro, macro, weighted), as shown in Fig. 5. We performed eight different runs and reported the average, standard deviation, and elapsed time. In general, sampling with a larger N tends to reduce the standard deviation and yield more reliable Markov Boundary estimates. Moreover, as we process more samples, the model gradually improves, following a logarithmic growth pattern until it converges to a final score. We also verify that our time complexity is linear with the number of samples N. Based on these results, we generally select $N = 68$ as the sample size.

C.3 Dynamic Thresholding

We performed ablations on the effect of k during the dynamic thresholding of the CMI (Eq. (9)) to access conditional independence in Fig. 6. To balance the classification metrics across the different averaging, we set $k = 2.75$.

Figure 5: Evolution of several classification metrics (one-shot) and elapsed time per sample in function of the number of samples N chosen. Results are reported using 1-sigma error bar.

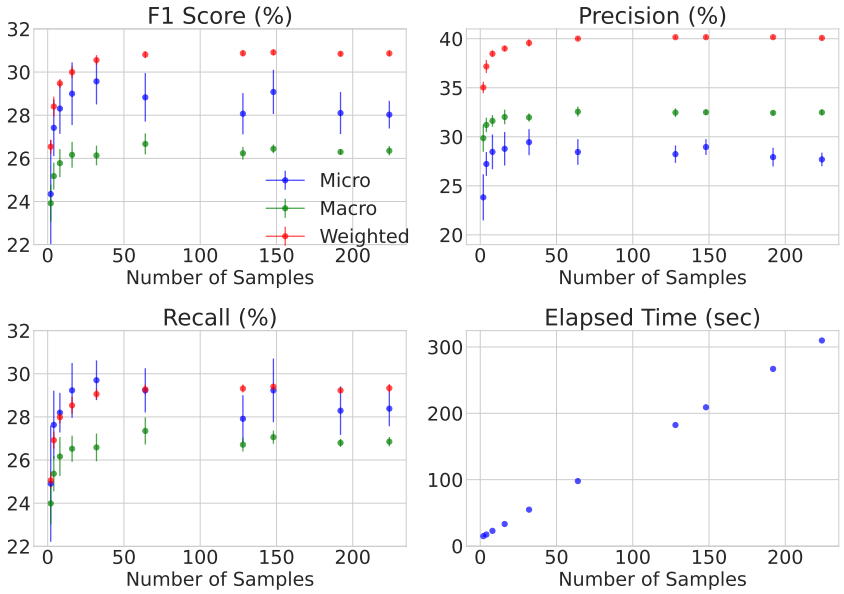
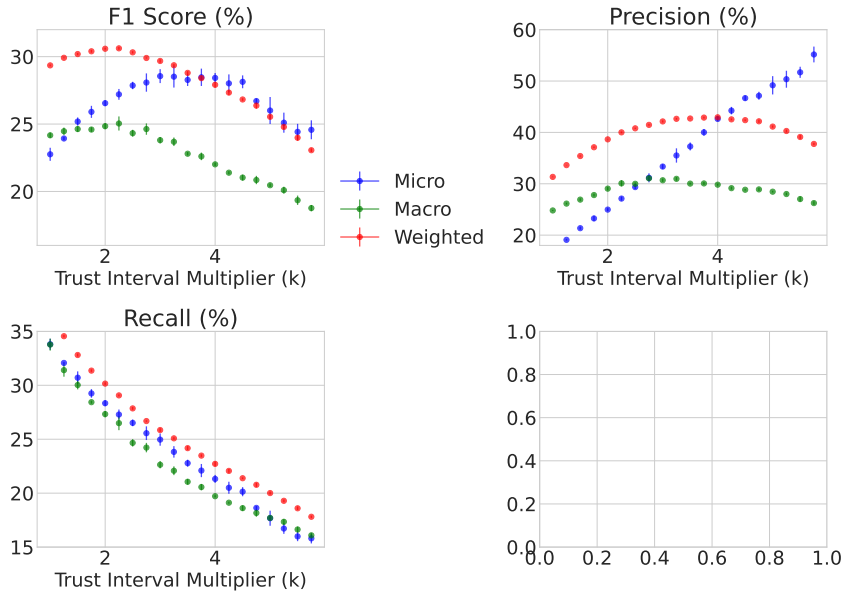


Figure 6: Evolution of one-shot F1 Score, Precision, and Recall in function of coefficient k. Results are reported using 1-sigma error bar.



C.4 Criterions

This section presents the different criteria used for comparison in the experimental evaluation.

C.5 Frequency

Frequency-based heuristics that apply fixed thresholds τ to the empirical frequency of the occurrence of X_i in each of the Markov Boundary $\mathbf{MB}(Y_j)$. Formally, For each label Y_j, after merging all local edge sets into a global set $E = \bigcup_{i=1}^{m} E_i$, we evaluate each candidate variable $X_i \in \mathbf{MB}_j$ based on

its frequency of appearance across the local models. If this frequency exceeds the threshold τ, the variable is retained in the final merged $\mathbf{MB}_j$; otherwise, it is discarded.

C.5.1 Expected FPR Adjustment

Same principle as in [8] except that we fit the two Beta distributions on the mutual information of $I(Y_j, X_i)$ instead of the raw frequencies.

C.5.2 Mutual Information

BES The BES is the second phase of GES [2], where edges are removed one after the other to maximize a criterion S. Heuristics approaches [35, 5] aim to solve this problem by optimizing:

$$E = \underset{E_i \in \varepsilon}{\arg\max} \sum_{e \in E_i} S(e) \tag{15}$$

Where ε denotes the search space (all possible edges over $\boldsymbol{U}$) and $S(e)$ a criterion function for edge relevance (e.g. edge frequency, thresholds, $\cdots$). This formulation takes into account the underlying edges' characteristics but not the overall network structure, complexity and missing data [32], leading to a *consensus fusion approach* [44].

Estimating Mutual Information in Event Sequences. We want to reuse the estimated conditional mutual information, Eq. (4), and profit from the parallelized inference of Phase 1 (Fig. 2).

As argued out by Janzing et al. [18], a causal strength measure (or criterion) $C_{X_i \to Y_j}$ should possess multiple properties. Notably, if $C_{X_i \to Y_j} = 0$, then the joint distribution satisfies the Markov condition with respect to the DAG obtained by removing the arrow $X_i \to Y_j$. Moreover, the true DAG reads $X_i \to Y$ iff $C_{X_i \to Y_j} = I(X_i, Y_j)$.

It is a natural criterion for merging edges across multiple causal graphs. However, it remains tricky to estimate [19]. Using the chain rule of conditional mutual information [3], we can rewrite it as:

$$I(Y_j, X_i | \boldsymbol{Z}) = I(Y_j, X_i) - I(Y_j, X_i, \boldsymbol{Z}) \tag{16}$$

Where $I(Y_j, X_i, \boldsymbol{Z})$ is the interaction information [4], which tells us whether knowing $\boldsymbol{Z}$ explains away the dependency between X_i and Y_j (negative interaction), or enhances it (positive interaction):

$$I(Y_j, X_i, \boldsymbol{Z}) \triangleq I(Y_j, \boldsymbol{Z}) - I(Y_j, \boldsymbol{Z} | X_i)$$

$I(Y_j, \boldsymbol{Z})$ can be estimated using the same Monte-Carlo sampling as for $I(Y_j, X_i | \boldsymbol{Z})$ (4). Since $I(Y_j, \boldsymbol{Z}) = H(Y_j) - H(Y | \boldsymbol{Z_j})$, the marginal $p(y)$ is needed. Fortunately, the dataset $\boldsymbol{D}$ is large enough, hence the frequencies of y_j are recovered empirically and an estimate $\hat{p}(y)$ which we assume to be equal to the true marginal $p(y)$. We acknowledge that under a restricted dataset, $\hat{p}(y)$ might differ from $p(y)$. This yields to:

$$I(Y_j, \boldsymbol{Z}) = \mathbb{E}_z D_{KL}(P(Y_j | \boldsymbol{Z}) || \hat{P}(Y_j)) = \mathbb{E}_z I_G(Y_j, z) \tag{17}$$

Formally, we assume that for long sequences i.e $i \to +\infty$, our event sequences form a stationary ergodic stochastic process and $I(Y_j, \boldsymbol{Z} | X_i)$ is negligible compare to $I(Y_j, \boldsymbol{Z})$ since $\boldsymbol{Z}$ is containing most of the information to predict Y_j. This reduces the mutual information to

$$I(Y_j, X_i) \approx I(Y_j, X_i | \boldsymbol{Z}) + I(Y_j | \boldsymbol{Z})$$

Criterion. We propose a **Class-Aware Information Gain (CAIG)** score for evaluating candidate edges during Phase 2 of CARGO. Given m i.i.d. samples from a dataset $\boldsymbol{D}$, CAIG balances three key factors: mutual information derived from information gain, class imbalance, and network complexity.

For each label node Y_j, with candidate parent set $\mathbf{Pa}_j^{\mathbb{G}}$, the CAIG score is:

$$S(\mathbb{G}', \boldsymbol{D}) = \sum_{j=1}^{n} s_I(Y_j, \mathbf{Pa}_j^{\mathbb{G}}) - \alpha \cdot |\mathbf{Pa}_j^{\mathbb{G}}| \cdot \log\left(\frac{m}{m_j} + 1\right) \tag{18}$$

With $s_I(Y_j, \mathbf{Pa}_j^{\mathbb{G}}) = \sum_{X_i \in \mathbf{Pa}_j^{\mathbb{G}}} I(Y_j, X_i)$, α is a regularization hyperparameter, m_j is the number of positive instances for class Y_j.

This formulation encourages informative yet parsimonious graph structures, correcting for underrepresented labels via the regularization term. It is also efficient since CAIG is decomposable [2] like BIC with the local s_I. This criterion is denoted as BES mi imbalance par in our experiments in Fig. 4.

D Implementation

D.1 Computation.

A key advantage of our approach is its scalability. Unlike traditional methods whose complexity depends on the event and label cardinality $|\mathbb{X}|$ and $|\mathbb{Y}|$ [23], our method is agnostic to both. As illustrated in Figure 2, all steps are parallelized on GPUs. CMI estimations are independently performed for all positions $i \in [c, L]$, with the sampling pushed into the batch dimension and results averaged across labels, leading to BS $\times N \times L$ CI-tests per batch $D = \{S_l^0, \ldots, S_l^m\}$. Consequently, time complexity transitions from $\mathcal{O}(\text{BS} \times N \times L)$ to $\mathcal{O}(1)$ per batch due to GPU parallelism. The complexity is bounded by the Transformers' inference part, where it scales quadratically with the sequence length $\mathcal{O}(L^2)$ if one uses vanilla self-attention [48].

D.2 Phase 1

The following is the implementation of the one-shot phase in PyTorch [31].

```
def topk_p_sampling(z, prob_x, c: int, n: int = 64, p: float = 0.8, k: int = 35,
                    cls_token_id: int = 1, temp: float = None):
    # Sample just the context
    input_ = prob_x[:, :c]

    # Top-k first
    topk_values, topk_indices = torch.topk(input_, k=k, dim=-1)

    # Top-p over top-k values
    sorted_probs, sorted_idx = torch.sort(topk_values, descending=True, dim=-1)
    cum_probs = torch.cumsum(sorted_probs, dim=-1)
    mask = cum_probs > p

    # Ensure at least one token is kept
    mask[..., 0] = 0

    # Mask and normalize
    filtered_probs = sorted_probs.masked_fill(mask, 0.0)
    filtered_probs += 1e-8  # for numerical stability
    filtered_probs /= filtered_probs.sum(dim=-1, keepdim=True)

    # Unscramble to match the original top-k indices
    # Need to reorder the sorted indices back to the original top-k
    reorder_idx = torch.argsort(sorted_idx, dim=-1)
    filtered_probs = torch.gather(filtered_probs, -1, reorder_idx)

    batched_probs = filtered_probs.unsqueeze(1).repeat(1, n, 1, 1)
        # (bs, n, seq_len, k)
    batched_indices = topk_indices.unsqueeze(1).repeat(1, n, 1, 1)
        # (bs, n, seq_len, k)

    sampled_idx = torch.multinomial(batched_probs.view(-1, k), 1)
        # (bs*n*seq_len, 1)
    sampled_idx = sampled_idx.view(-1, n, c).unsqueeze(-1)

```

```
    sampled_tokens = torch.gather(batched_indices, -1, sampled_idx).squeeze(-1)
    sampled_tokens[..., 0] = cls_token_id

    # Reconstruct full sequence
    z_expanded = z.unsqueeze(1).repeat(1, n, 1)[..., c:]
    return torch.cat((sampled_tokens, z_expanded), dim=-1)

from torch import nn
def OneShotCD(tfe: nn.Module, tfy: nn.Module, batch: dict[str, torch.Tensor], c: int, n: int, eps: float=1e-6, topk: int=20, k: int=2.75, p=0.8) -> torch.Tensor:
    """ tfe, tfy: are the two autoregressive transformers (event type and label)
        batch: dictionary containing a batch of input_ids and attention_mask of shape (bs, L) to explain.
        c: scalar number defining the minimum context to start inferring, also the sampling interval.
        n: scalar number representing the number of samples for the sampling method.
        eps: float for numerical stability
        topk: The number of top-k most probable tokens to keep for sampling
        k: Number of standard deviations to add to the mean for dynamic threshold calculation
        p: Probability mass for top-p nucleus
    """
    o = tfe(attention_mask=batch['attention_mask'], input_ids=batch['input_ids'])['prediction_logits'] # Infer the next event type
    x_hat = torch.nn.functional.softmax(o, dim=-1)

    b_sampled = topk_p_sampling(batch['input_ids'], x_hat, c, k=topk, n=n, p=p) # Sampling up to (bs, n, L)
    n_att_mask = batch['attention_mask'].unsqueeze(1).repeat(1, n, 1)

    with torch.inference_mode():
        o = tfy(attention_mask=n_att_mask.reshape(-1, b_sampled.size(-1)), input_ids=b_sampled.reshape(-1, b_sampled.size(-1))) # flatten and infer
        prob_y_sampled = o['ep_prediction'].reshape(b_sampled.size(0), n, batch['input_ids'].size(-1)-c, -1) # reshape to (bs, n, L-c)

        # Ensure probs are within (eps, 1-eps)
        prob_y_sampled = torch.clamp(prob_y_sampled, eps, 1 - eps)

        y_hat_i = prob_y_sampled[..., :-1, :] # P(Yj|z)
        y_hat_iplus1 = prob_y_sampled[..., 1:, :] # P(Yj|z, x_i)

        # Compute the CMI & CS and average across sampling dim
        cmi = torch.mean(y_hat_iplus1*torch.log(y_hat_iplus1/y_hat_i)+(1-y_hat_iplus1)*torch.log((1-y_hat_iplus1)/(1-y_hat_i)), dim=1)
        # (BS, L, Y)
        cs = y_hat_iplus1 - y_hat_i
        cs_mean = torch.mean(cs, dim=1)
        cs_std = torch.std(cs, dim=1)

        # Confidence interval for threshold
        mu = cmi.mean(dim=1)
        std = cmi.std(dim=1)
        dynamic_thresholds = mu + std * k

        # Broadcast to select an individual dynamic threshold
        cmi_mask = cmi >= dynamic_thresholds.unsqueeze(1)

        cause_token_indices = cmi_mask.nonzero(as_tuple=False)
```

```
        # (num_causes, 3) --> each row is [batch_idx, position_idx, label_idx]
        return cause_token_indices, cs_mean, cs_std, cmi_mask
```

Remark. *Since tfy contains tfe as backbone, in practice we need only one forward pass from tfy and extract also $\hat{x}$, so tfe is not needed. We let it to improve understanding and clarity.*

D.3 Phase 2.

```
import random
def create_auto_adaptive_threshold_fn(all_m_j, tau_max=0.5, tau_min=0.05, k=None, m0="median"):
    m_0 = np.median(all_m_j)

    if k == None:
        q25, q75 = np.percentile(all_m_j, [25, 75])
        if q75 == q25:
            k = 1.0
        else:
            log_iqr = np.log(q75) - np.log(q25)
            k = (2 * np.log(3)) / log_iqr

    def threshold_function(m_j):
        log_m_j = np.log(m_j + 1e-9)
        log_m_0 = np.log(m_0)
        logistic_decay = 1 / (1 + np.exp(k * (log_m_j - log_m_0)))
        return (tau_max - tau_min) * logistic_decay + tau_min

    return threshold_function

def adaptive_thresholding_frequency(graphs: list,
                present_labels: dict,
                frequency_threshold: float = 0.5,
                k: float=None,
                tau_min: float=0.05,
                tau_max: float=0.5,
                m0: str="median",
                verbose=False,
                **kwargs):
    """
    Frequency voting: keep edges appearing with frequency > threshold across samples.

    :param graphs: list of local graphs (e.g., from Phase 1). Each graph is a dict[label][token] = list of stats.
    :param present_labels: labels present in evaluation
    :param frequency_threshold: e.g. 0.5 for majority, 0.8 for conservative
    :return: filtered_labels, sample_per_label, elapsed_time
    """
    start_time = datetime.now()
    # Step 1: Aggregate graphs
    labels, sample_per_label = union(graphs)  # user-defined union function
    old_labels = labels.copy()
    nodes = count_nodes(labels)
    samples = len(graphs)

    # Create threshold function
    auto_threshold_fn = create_auto_adaptive_threshold_fn(list(sample_per_label.values()), k=k, tau_max=tau_max, tau_min=tau_min, m0=m0)

    # Step 2: Frequency voting with dynamic thresholds
```

```
    edge_counts = defaultdict(lambda: defaultdict(int))  # edge_counts
[label][token] = count

    for g in graphs:
        for label, token_dict in g.items():
            if label not in labels:
                continue
            for token in token_dict:
                edge_counts[label][token] += 1

    # Step 3: Keep edges above frequency threshold
    filtered_labels = defaultdict(dict)
    for label in labels:
        total = sample_per_label.get(label, samples)  # fallback to
total graphs if missing
        for token, count in edge_counts[label].items():
            freq = count / total
            if freq >= auto_threshold_fn(sample_per_label.get(label,
1)):
                filtered_labels[label][token] = {'frequency': freq}
                if verbose:
                    print(f"[{label}] token {token} kept (freq={freq
:.2f})")

    nb_of_edges = sum(len(v) for v in filtered_labels.values())
    print(f"Time: {(datetime.now() - start_time).total_seconds():.2f}s
")
    return filtered_labels, sample_per_label, (datetime.now() -
start_time).total_seconds()
```

E Figures

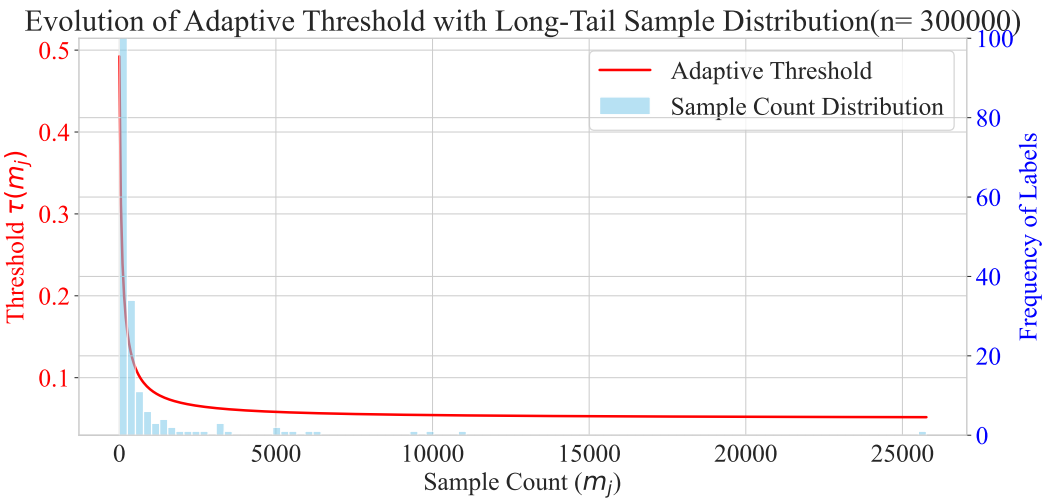


Figure 7: Adaptive thresholding function $\tau_j(m_j)$ across varying label frequencies m_j, illustrating the logistic decay from $\tau_{\max}$ to $\tau_{\min}$.

Figure 8: Illustration of structural fusion: individual causal graphs (left) aggregated into a fused DAG for multi-label event sequences (right) using a simple union.