Anytime Approximate Formal Feature Attribution

Jinqiang Yu[1,2], Graham Farr[1], Alexey Ignatiev[1], Peter J. Stuckey[1,2]

1. Department of Data Science and AI, Monash University, Australia
2. Australian Research Council OPTIMA ITTC, Australia

Table of Contents

Explainable Artificial Intelligence (XAI)

Rapid advances in Artificial Intelligence (AI) and Machine Learning (ML) algorithms.

Issue: opaque models $\rightarrow$ lack of trust.

Rise: Explainable Artificial Intelligence (XAI).

Solution:Post-hoc explanations.

Post-hoc Explanations

Post-hoc explanations answer '*why*?' questions and '*how*?'questions

Heuristic approach:

- Feature selection: Anchor.
- Feature Attribution: LIME, SHAP, etc.
- Issue:
  - Explanation quality.

Post-hoc Explanations

Post-hoc explanations answer '*why*?' questions and '*how*?'questions

Formal approach:

- Correct and minimal.
- Feature selection: abductive explanation (AXp) and contrastive explanation (CXp).
- Feature attribution: formal feature attribution (FFA).

FFA

Formal feature attribution (FFA):

Provide the importance of each feature, i.e. the proportion of AXp's in which it appears.

FFA approach:

- Make use of the *hitting set duality* between AXp's and CXp's.
- Collect AXp's as a *side effect* of CXp enumeration algorithm.

FFA Observations

Observations from the FFA approach:

- Usually find many AXp's before finding the first CXp.
- AXp's are diverse $\rightarrow$ good approximation of FFA.
- AXp enumeration can get *exact* FFA faster.

Issue: *Exact* FFA is hard to compute.

Table of Contents

SAT

Boolean Satisfiability (SAT)

- Decision problem for propositional logic.
- Formula φ: Conjunctive normal form (CNF).
  - Clause: a disjunction of literals.
  - Literal: a Boolean variable b or $\neg b$.
  - Example: $(a \vee \neg c) \wedge (b \vee c)$.
- Satisfiable: there exists an assignment μ satisfying the formula.

MaxSAT

Maximum Satisfiability (MaxSAT): maximize the number of satisfied clauses.

Partial Unweighted MaxSAT: $\phi = \mathcal{H} \wedge \mathcal{S}$.

- $\mathcal{H}$: hard clauses, which *must* be satisfied.
- $\mathcal{S}$: soft clauses, which represent a *preference* to satisfy those clauses.
- Aim: maximize the number of satisfied soft clauses.

MUS and MCS

Let $\phi = \mathcal{H} \cup \mathcal{S}$ and $\phi \models \perp$.

Minimal Unsatisfiable Subset (MUS):

A subset of clauses $\mu \subseteq \mathcal{S}$ is a *MUS* iff $\mathcal{H} \cup \mu \models \perp$ and $\forall \mu' \subsetneq \mu$ it holds that $\mathcal{H} \cup \mu' \not\models \perp$.

Minimal Correction Subset (MCS):

A subset of clauses $\sigma \subseteq \mathcal{S}$ is a *MCS* iff $\mathcal{H} \cup \mathcal{S} \setminus \sigma \not\models \perp$ and $\forall \sigma' \subsetneq \sigma$ it holds that $\mathcal{H} \cup \mathcal{S} \setminus \sigma' \models \perp$.

Minimal Hitting Set Duality

Minimal hitting set (MHS) duality relationship between MUSes and MCSes, i.e.

$\mathbb{U}_\phi = \mathrm{MHS}(\mathbb{C}_\phi)$ and $\mathbb{C}_\phi = \mathrm{MHS}(\mathbb{U}_\phi)$

where

- $\mathbb{U}_\phi$: MUSes.
- $\mathbb{C}_\phi$: MCSes.
- $\mathrm{MHS}(S)$ returns the minimal hitting sets of S.
- *Minimal* sets that share an element with each subset in S.

Classification Example

A classification function τ: $\mathcal{F} \rightarrow K$.

- $\mathcal{F}$: complete feature space.
- K: a set of classes.

Classification Problem

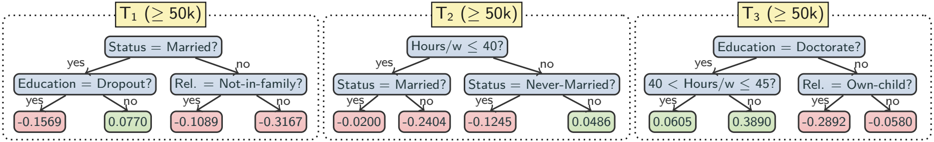


Figure 1: Boosted tree model trained on the *adult* classification dataset.

Instance:

{Education=Bachelors, Status=Separated, Occupation=Sales, Relationship=Not-in-family, Sex=Male, Hours/w≤40}

Score: $-0.4073 = (-0.1089 - 0.2404 - 0.0580) < 0 \rightarrow$ prediction $< 50k$

Formal Explanation

Abductive explanation (AXp) $\mathcal{X}$: subset-minimal set of features sufficing to explain the prediction .

$$\forall(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{i\in\mathcal{X}}(x_i = v_i)\right] \rightarrow (\kappa(\mathbf{x}) = c)$$

Contrastive explanation (CXp) $\mathcal{Y}$: subset-minimal set of features that are necessary to change the prediction.

$$\exists(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{i\notin\mathcal{Y}}(x_i = v_i)\right] \wedge (\kappa(\mathbf{x}) \neq c)$$

Minimal Hitting Set Duality: CXps minimally hit every AXp, and vice-versa.

AXp and CXp Examples

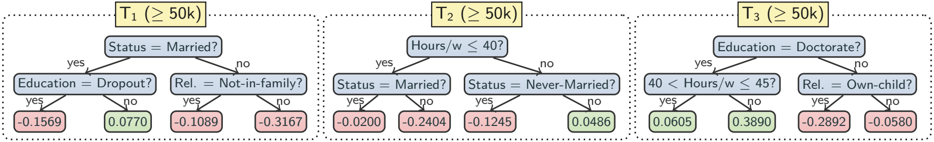


Figure 2: Boosted tree model trained on the *adult* classification dataset.

$\mathcal{X}_1 = \{\,\text{Education, Hours/w}\,\}$	
IF	Education = Bachelors
AND	Hours/w $\leq$ 40
THEN	Target $<50k$

(a) AXp $\mathcal{X}_1$.

$\mathcal{X}_2 = \{\,\text{Education, Status}\,\}$	
IF	Education = Bachelors
AND	Status = Separated
THEN	Target $<50k$

(b) AXp $\mathcal{X}_2$.

AXp and CXp Examples

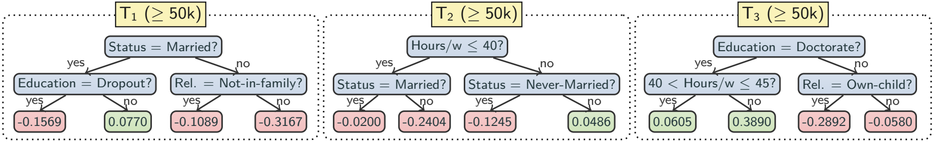


Figure 2: Boosted tree model trained on the *adult* classification dataset.

$\mathcal{Y}_1 = \{\text{Education}\}$	
IF	Education $\neq$ Bachelors
THEN	Target can be changed to $\geq 50k$

(a) CXp $\mathcal{Y}_1$.

$\mathcal{Y}_2 = \{\text{Hours/w, Status}\}$	
IF	Hours/w $\not\leq$ 40
AND	Status $\neq$ Separated
THEN	Target can be changed to $\geq 50k$

(b) CXp $\mathcal{Y}_2$.

Formal Feature Attribution (FFA)

Inspired by the implicit hitting set based algorithm eMUS/MARCO.

$$\mathrm{ffa}_\kappa(i, (\mathbf{v}, c)) = \frac{|\{\mathcal{X} \mid \mathcal{X} \in \mathbb{A}_\kappa(\mathbf{v}, c), i \in \mathcal{X})|}{|\mathbb{A}_\kappa(\mathbf{v}, c)|} \tag{1}$$

where

- $\mathbf{v}$: instance.
- c: prediction.
- $\mathcal{X}$: AXp.
- $\mathbb{A}_\kappa(\mathbf{v}, c)$: the set of AXp's for $\mathbf{v}$.
- i: feature.

FFA Algorithm

Algorithm 1 Anytime Explanation Enumeration

Input: Classifier: κ, instance: $\mathbf{v}$, prediction: c

Output: AXp's: $\mathbb{A}$, CXp's: $\mathbb{C}$

$(\mathbb{A}, \mathbb{C}) \leftarrow (\emptyset, \emptyset)$ $\triangleright$ *Sets of AXp's and CXp's to collect.*

while resources available **do**

  $\mathcal{Y} \leftarrow \textsc{MinimalHS}(\mathbb{A}, \mathbb{C})$ $\triangleright$ *Get a new MHS of $\mathbb{A}$ subject to $\mathbb{C}$.*

  $\triangleright$ $\mathcal{Y}$ is the CXp candidate.

  if $\mathcal{Y} = \perp$ **then break** $\triangleright$ *Stop if none is computed.*

  $\triangleright$ Check CXp condition for $\mathcal{Y}$.

  if $\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{i \notin \mathcal{Y}} (x_i = v_i)\right] \wedge (\kappa(\mathbf{x}) \neq c)$ **then**

    $\mathbb{C} \leftarrow \mathbb{C} \cup \{\mathcal{Y}\}$ $\triangleright$ *$\mathcal{Y}$ appears to be a CXp.*

  else

    $\mathcal{X} \leftarrow \textsc{ExtractAXp}(\mathcal{F} \setminus \mathcal{Y}, \kappa, \mathbf{v}, c)$

    $\mathbb{A} \leftarrow \mathbb{A} \cup \{\mathcal{X}\}$ $\triangleright$ *There must be a missing AXp $\mathcal{X} \subseteq \mathcal{F} \setminus \mathcal{Y}$.*

return $\mathbb{A}$, $\mathbb{C}$

Algorithm 2 ExtractAXp

Input: Candidate: $\mathcal{X}$, classifier: κ, instance: $\mathbf{v}$, prediction: c

Output: AXp: $\mathcal{X}$

for $j \in \mathcal{X}$ **do**

  if $\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{i \in \mathcal{X} \setminus \{j\}} (x_i = v_i)\right] \rightarrow (\kappa(\mathbf{x}) = c)$ **then**

    $\mathcal{X} \leftarrow \mathcal{X} \setminus \{j\}$

return $\mathcal{X}$

FFA Example

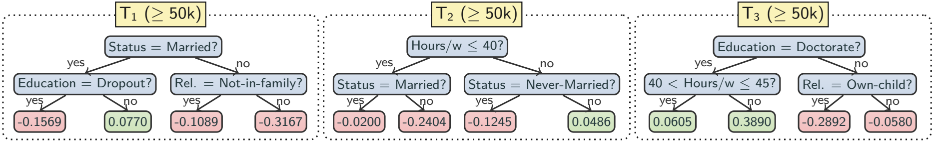


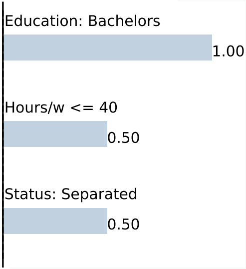
Figure 4: Boosted tree model trained on the *adult* classification dataset.

$\mathcal{X}_1 = \{\text{Education, Hours/w}\}$	
IF	Education = Bachelors
AND	Hours/w $\leq$ 40
THEN	Target $<$50k

(a) AXp $\mathcal{X}_1$.

$\mathcal{X}_2 = \{\text{Education, Status}\}$	
IF	Education = Bachelors
AND	Status = Separated
THEN	Target $<$50k

(b) AXp $\mathcal{X}_2$.



(c) FFA.

LIME, SHAP and FFA Examples

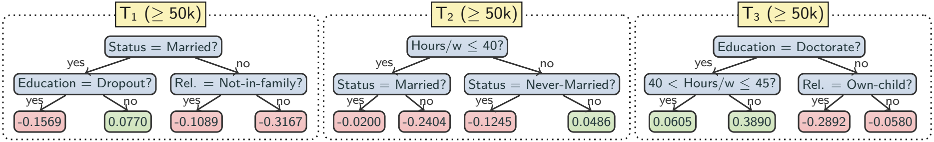
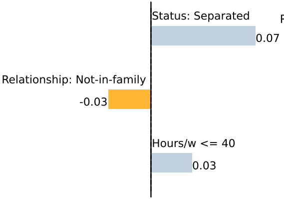
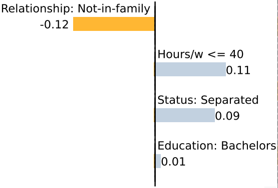


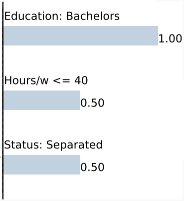
Figure 6: Boosted tree model trained on the *adult* classification dataset.



(a) LIME.



(b) SHAP.



(c) FFA.

Issue of LIME and SHAP: Some *irrelevant* features have non-zero attribution, e.g., *relationship*

Table of Contents

Switching from CXp to AXp Enumeration

Issue in the FFA approach: *Exact* FFA hard to compute.

Inspirations:

- AXp enumeration $\rightarrow$ getting *exact* FFA faster.
- Diverse AXp's $\rightarrow$ good FFA approximations.
- AXp enumeration $\rightarrow$ AXp's are not diverse.
- CXp enumeration $\rightarrow$ diverse AXp's $\rightarrow$ quick convergence.

Proposed approach:

- Anytime approach to computing approximate FFA
- Start with CXp enumeration
- Switch to AXp enumeration at some point.

Algorithm with Switching

Algorithm 3 Adaptive Explanation Enumeration

$(\mathbb{E}_0, \mathbb{E}_1) \leftarrow (\emptyset, \emptyset)$ ▷ *CXp's and AXp's to collect*
$\rho \leftarrow 0$ ▷ *Target phase of enumerator, initially CXp*
while true **do**
  $\mu \leftarrow \textsc{MinimalHS}(\mathbb{E}_{1-\rho}, \mathbb{E}_\rho, \rho)$ ▷ *MHS of $\mathbb{E}_{1-\rho}$ s.t. $\mathbb{E}_\rho$.*
  if $\mu = \bot$ **then break** ▷ *Stop if none is computed.*
  ▷ Check CXp condition for $\mathcal{Y}$.
  if $\textsc{IsTargetXp}(\mu, \tau, \mathbf{v}, c)$ **then**
    $\mathbb{E}_\rho \leftarrow \mathbb{E}_\rho \cup \{\mu\}$ ▷ *There must be a missing AXp $\mathcal{X} \subseteq \mathcal{F} \setminus \mathcal{Y}$.*
  else
    ▷ Collect target expl. μ
    $\nu \leftarrow \textsc{ExtractDualXp}(\mathcal{F} \setminus \mu, \tau, \mathbf{v}, c)$
    $\mathbb{E}_{1-\rho} \leftarrow \mathbb{E}_{1-\rho} \cup \{\nu\}$ ▷ *Collect dual expl. ν*
  ▷ The difference!
  if $\textsc{IsSwitchNeeded}(\mathbb{E}_\rho, \mathbb{E}_{1-\rho}, w, \alpha, \varepsilon)$ **then**
    $\rho \leftarrow 1 - \rho$ ▷ *Flip phase of MinimalHS*
return $\mathbb{E}_1$, $\mathbb{E}_0$ ▷ *Result AXp's and CXp's*

Switching Criteria

Criterion 1: Switch when CXp's on average are *much* smaller than AXp's, i.e. when

$$\frac{\sum_{\mathcal{X} \in \mathbb{A}^w} |\mathcal{X}|}{\sum_{\mathcal{Y} \in \mathbb{C}^w} |\mathcal{Y}|} \geq \alpha, \tag{2}$$

Criterion 2: Switch when the average CXp size "stabilizes".

$$\left| |\mathcal{Y}_{\text{new}}| - \frac{\sum_{\mathcal{Y} \in \mathbb{C}^w} |\mathcal{Y}|}{w} \right| \leq \varepsilon, \tag{3}$$

Rational:

- Normally $|\mathcal{X}| > |\mathcal{Y}|$
- CXp extraction: check satisfiable $\rightarrow$ cheap.
- AXp extraction: check unsatisfiable $\rightarrow$ expensive.
- Before switching: ensure AXp's diverse.
- After switching: single call for AXp, multiple calls for CXp extraction.

Switching Criteria

Criterion 1: Switch when CXp's on average are *much* smaller than AXp's, i.e. when

$$\frac{\sum_{\mathcal{X} \in \mathbb{A}^w} |\mathcal{X}|}{\sum_{\mathcal{Y} \in \mathbb{C}^w} |\mathcal{Y}|} \geq \alpha, \tag{2}$$

Criterion 2: Switch when the average CXp size "stabilizes".

$$\left| |\mathcal{Y}_{\text{new}}| - \frac{\sum_{\mathcal{Y} \in \mathbb{C}^w} |\mathcal{Y}|}{w} \right| \leq \varepsilon, \tag{3}$$

Switch when meeting *either* of the two criteria!

Table of Contents

Experimental Setup

Datasets: 3 Images and 2 text data

Metrics:

- **Errors**: Manhattan distance, i.e. the sum of absolute differences across all features.
- **Kendall's Tau**: Similarity of two rankings. Ranging $[-1, 1]$. The higher the closer.
- **Rank-biased overlap (RBO)**: Similarity of two rankings. Ranging $[0, 1]$. The higher the closer.
- **Kullback–Leibler (KL) divergence**: Statistical distance between two probability distributions. Ranging from 0 to ∞.
- **Number of AXp's**

Runtime

Average Runtime:

- MARCO-S (Our approach): 3509.50s (9.26s – 30881.55s)
- MARCO-A (AXp enumeration): 3255.30s (2.15s – 29191.42s)
- MARCO-C (CXp enumeration): 19311.87s (9.39s – 55951.57s)

Error Results

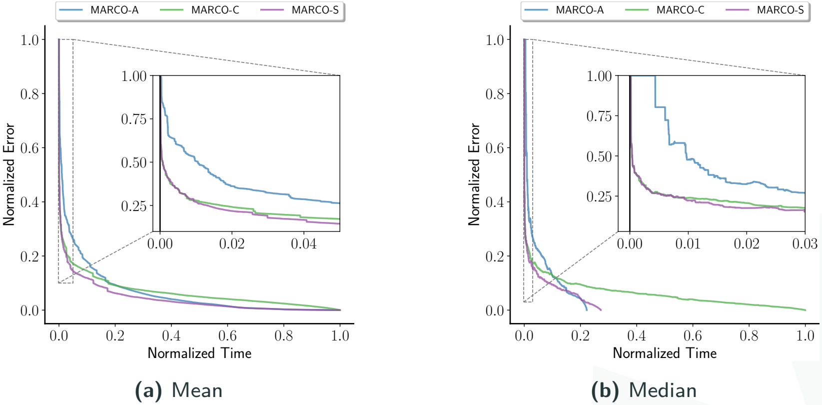
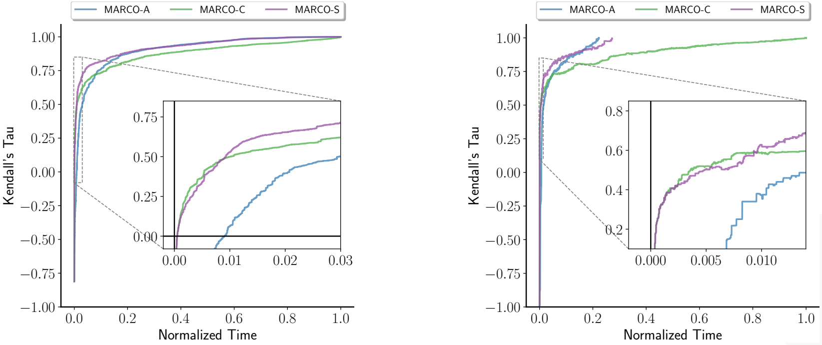


Figure 8: FFA approximation error over time.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

Kendall's Tau Results

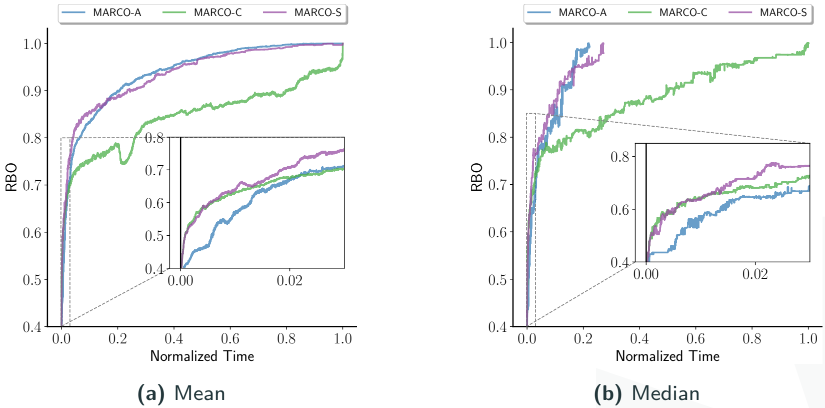


(a) Mean **(b)** Median

Figure 9: Kendall's Tau over time.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

RBO Results

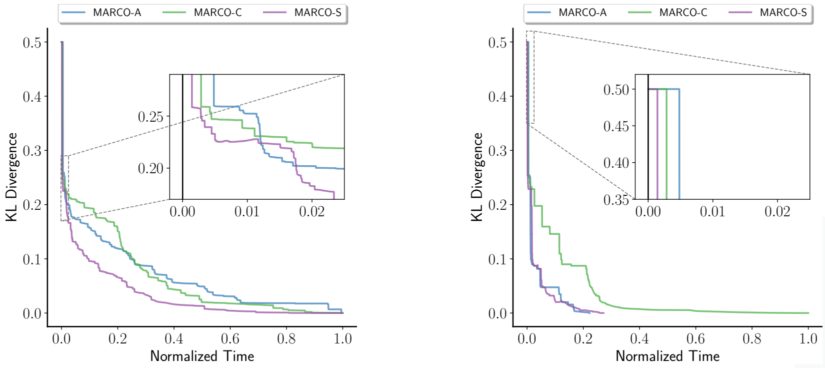


(a) Mean **(b)** Median

Figure 10: RBO over time.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

KL Divergence Results

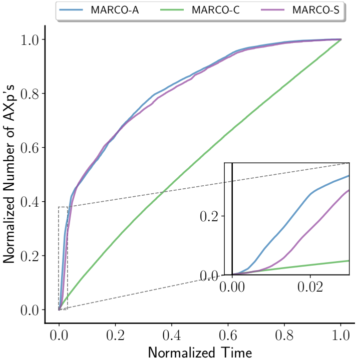


(a) Mean (b) Median

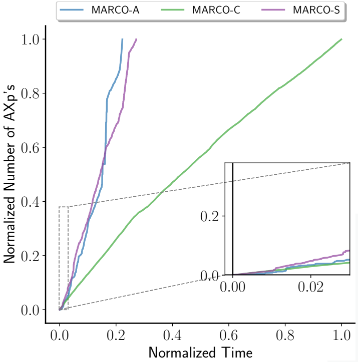
Figure 11: KL divergence over time.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

Number of AXp's Results



(a) Mean

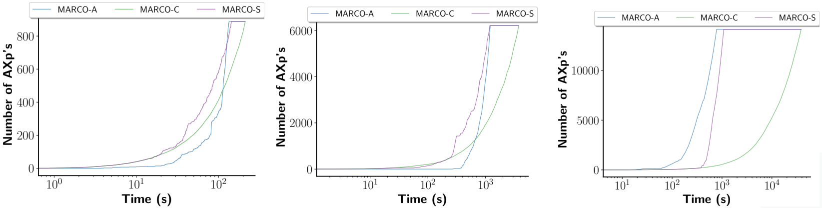


(b) Median

Figure 12: Number of AXp's over time.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

Number of AXp's Examples



(a) MNIST-1vs7 **(b)** PneumoniaMNIST **(c)** Sarcasm

Figure 13: Number of AXp's over time in example instances.

- MARCO-S: Propose approach
- MARCO-A: AXp enumeration
- MARCO-C: CXp enumeration

Table of Contents

Conclusions

- The proposed approach can replicate the behavior of the superior competitor $\rightarrow$ efficient and good approximation of FFA.
- Start with CXp enumeration $\rightarrow$ diverse AXp's.
- Switching to AXp enumeration $\rightarrow$ extracting AXp's faster.
- The proposed mechanism can be readily adapted to a multitude of other problems, e.g. in over-constrained systems or model-based diagnosis (MBD)

Thank you!