Honey or Poison? Solving the Trigger Curse in Few-shot Event Detection via Causal Intervention

Jiawei Chen[1,3], Hongyu Lin[1], Xianpei Han[1,2], Le Sun[1,2]

[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science

Institute of Software, Chinese Academy of Sciences

[3]University of Chinese Academy of Sciences

Outline

- Introduction
- Causal Intervention for Trigger Curse
- Experiments
- Conclusion

Outline

- Introduction
- Causal Intervention for Trigger Curse
- Experiments
- Conclusion

Event Detection

- Event detection aims to identify and classify event triggers in text.

Event Detection

- Event detection aims to identify and classify event triggers in text.

They were **killed** by hostile **fire** in Iraqi.

Die *Attack*

Few-shot Event Detection

- Supervised event detection require a large number of annotated data.
- Few-shot event detection (FSED) aims to detect new event with a few labeled data.

Few-shot Event Detection

- Supervised event detection require a large number of annotated data.

- Few-shot event detection (FSED) aims to detect new event with a few labeled data.

- **Support set**
  - A few labeled data
- **Query**
  - Unlabeled data

FSED System →

Detected events in **Query**

Trigger curse for few-shot event detection

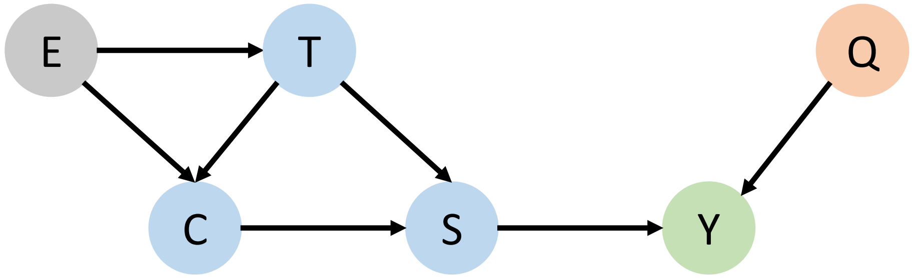
- Trigger curse
  - Overfitting the trigger will harm the generalization ability
  - Underfitting it will hurt the detection performance

Trigger curse for few-shot event detection

- Trigger curse
  - Overfitting the trigger will harm the generalization ability
  - Underfitting it will hurt the detection performance

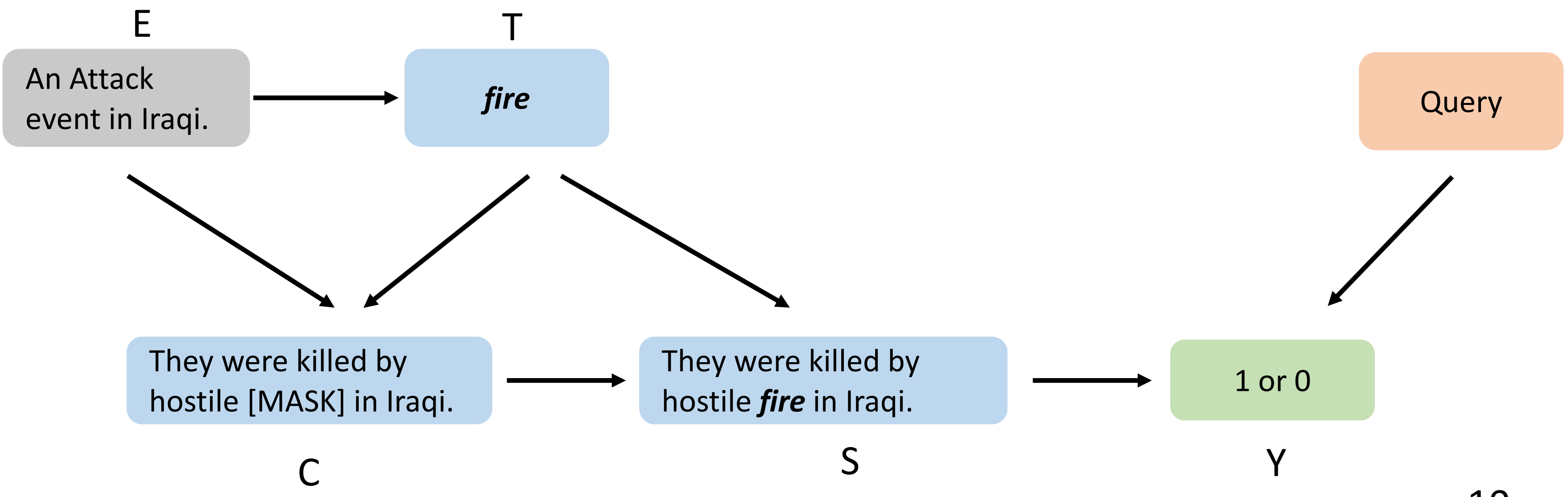
- For FSED, the support instances are very sparse and lack diversity.

Identify trigger curse from a causal view



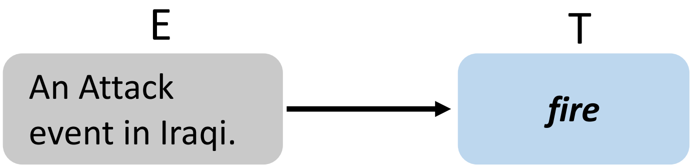
Structural Causal Model for few-shot event detection

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result



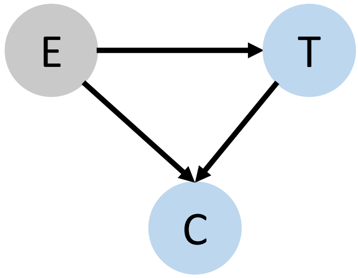
Structural Causal Model for FSED

$E \rightarrow T$: Trigger decision process



E : Event
T : Trigger set of support set

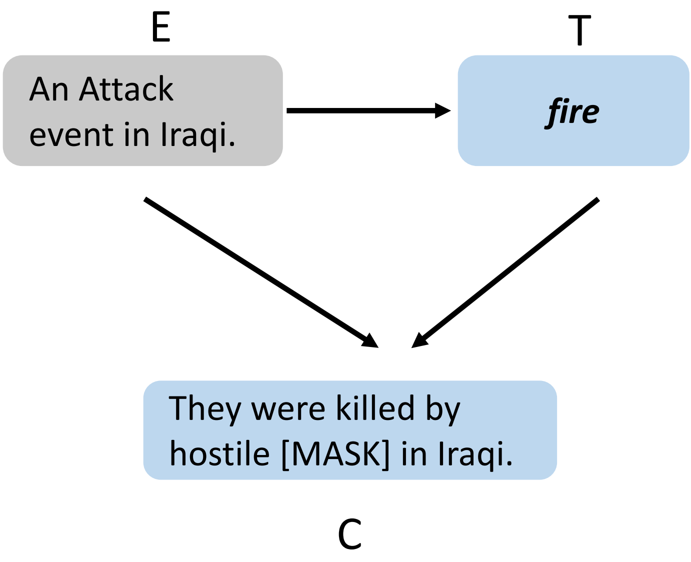
Structural Causal Model for FSED



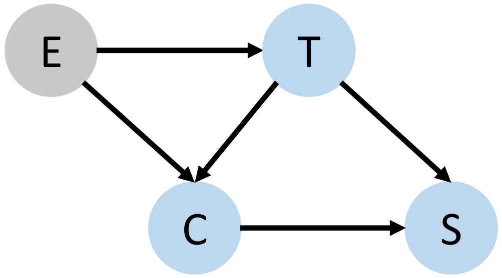
$E \rightarrow T$: Trigger decision process

$E \rightarrow C \leftarrow T$: Context is generated depending on both the event and the trigger

E : Event
T : Trigger set of support set
C : Context set of support set



Structural Causal Model for FSED

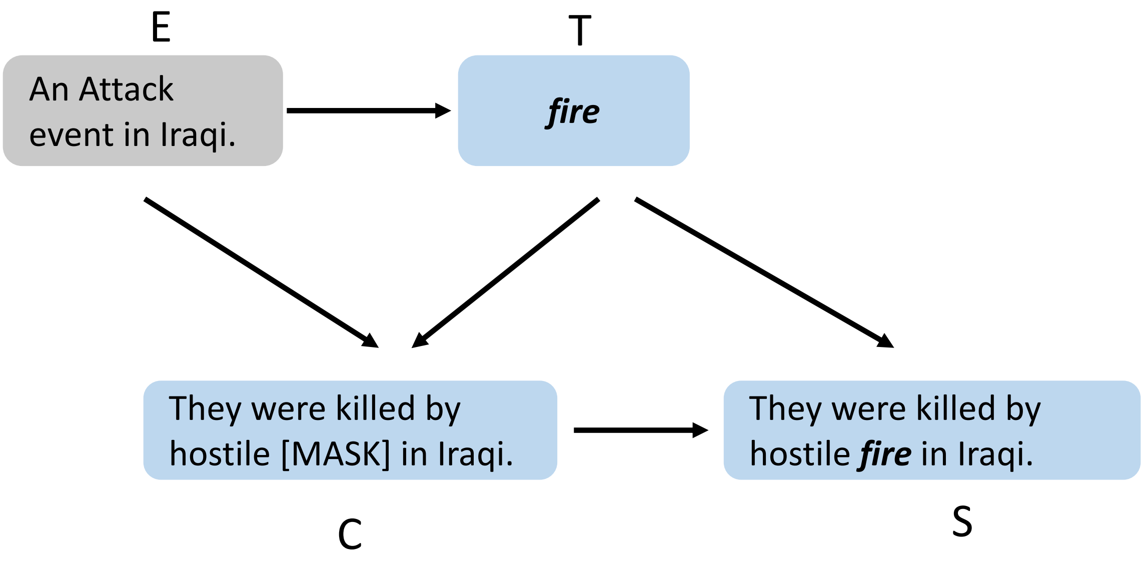


$E \rightarrow T$: Trigger decision process

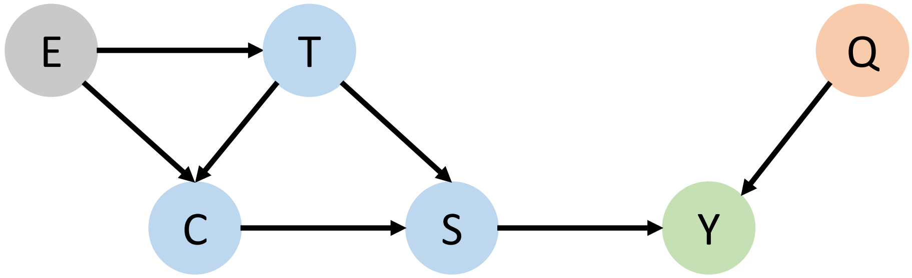
$E \rightarrow C \leftarrow T$: Context is generated depending on both the event and the trigger

$C \rightarrow S \leftarrow T$: Event instances are generated by combining context and the trigger.

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances



Structural Causal Model for FSED



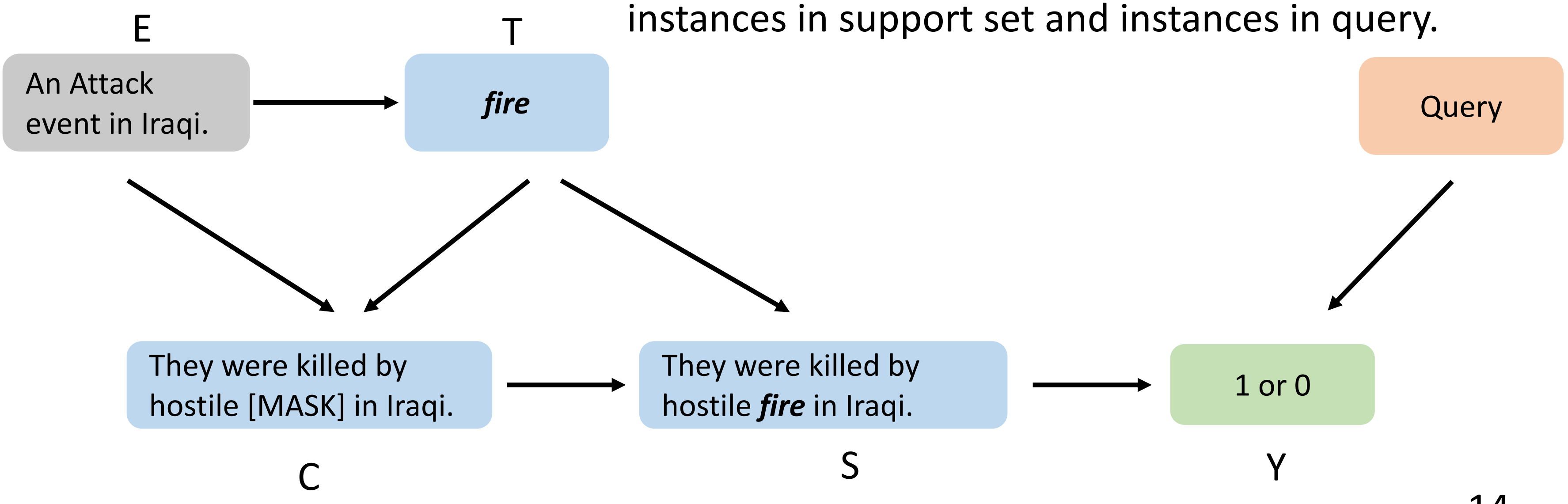
$E \rightarrow T$: Trigger decision process

$E \rightarrow C \leftarrow T$: Context is generated depending on both the event and the trigger

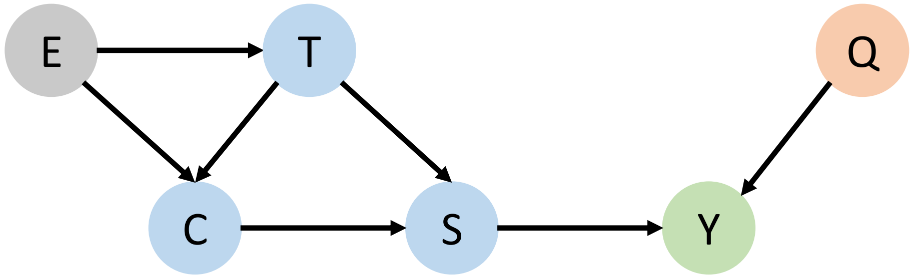
$C \rightarrow S \leftarrow T$: Event instances are generated by combining context and the trigger.

S$\rightarrow Y \leftarrow Q$: The result is predicted by the instances in support set and instances in query.

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result

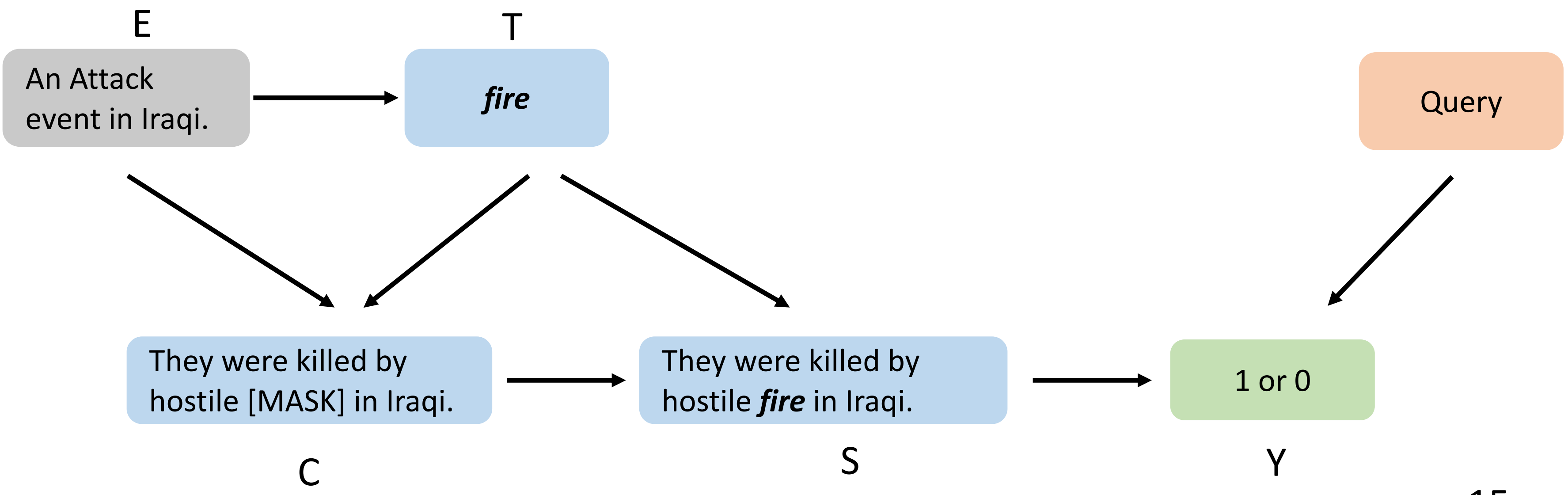


Structural Causal Model for FSED

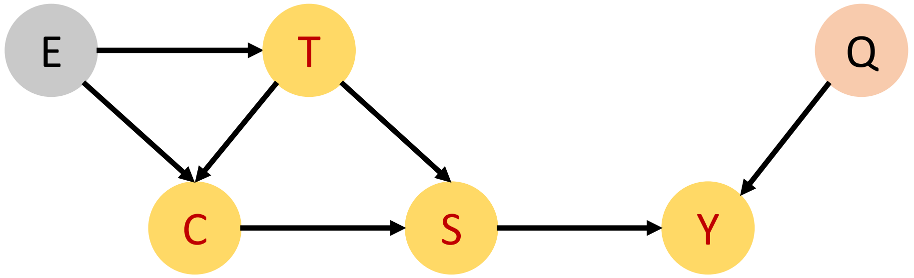


- Conventional learning criteria
  - $P(Y|S = s, Q = q)$

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result

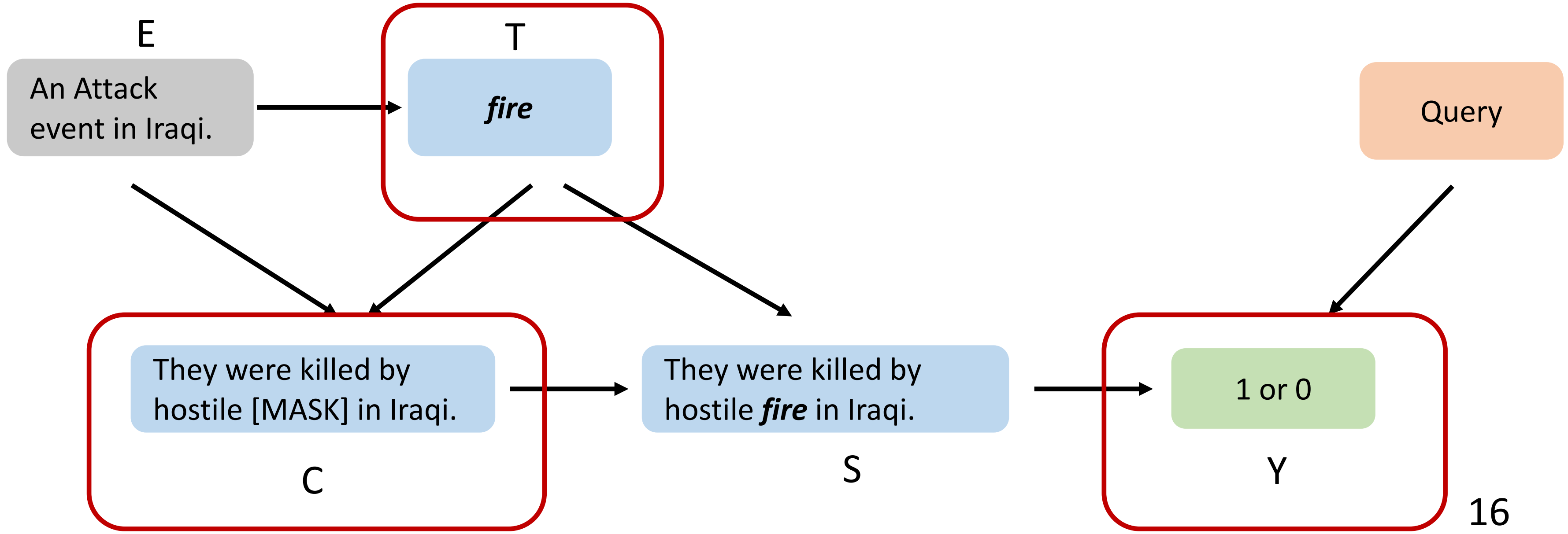


Structural Causal Model for FSED

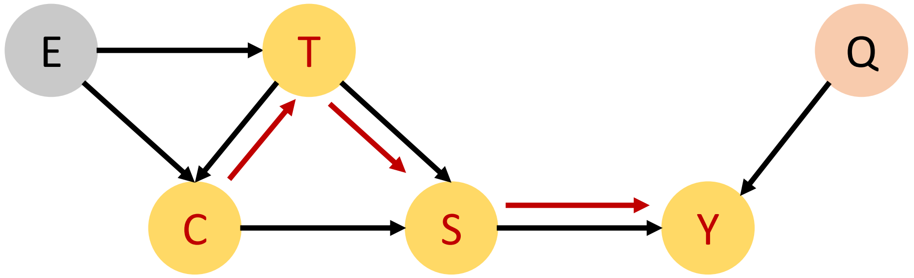


- Trigger is the confounder of context and predicted result.

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result

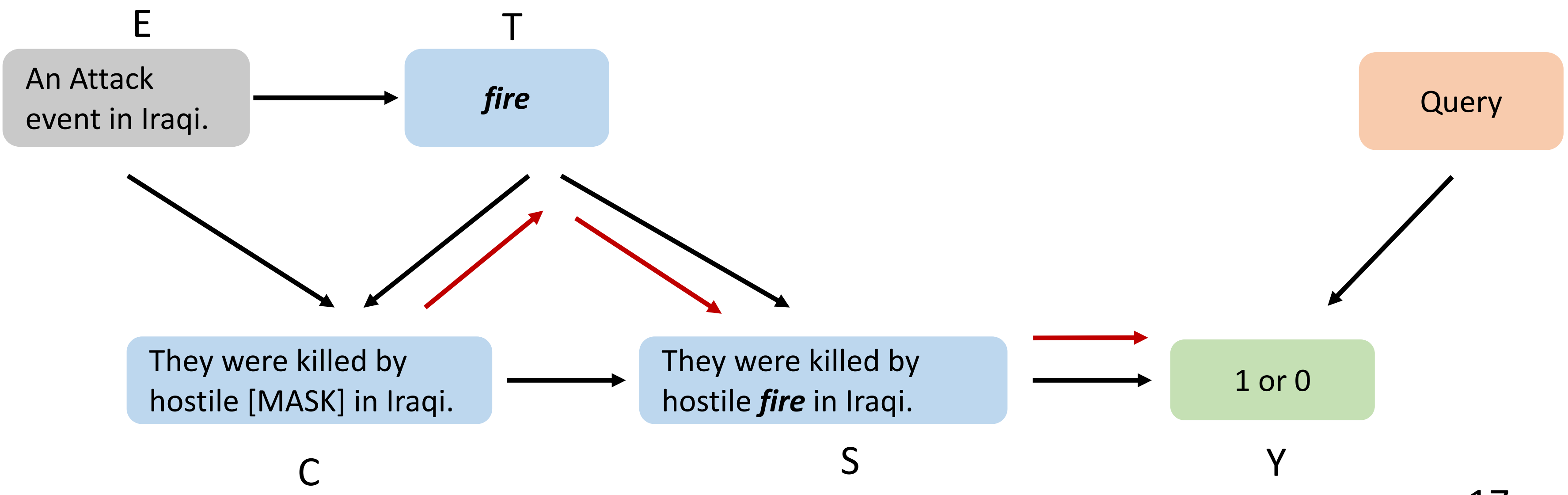


Structural Causal Model for FSED

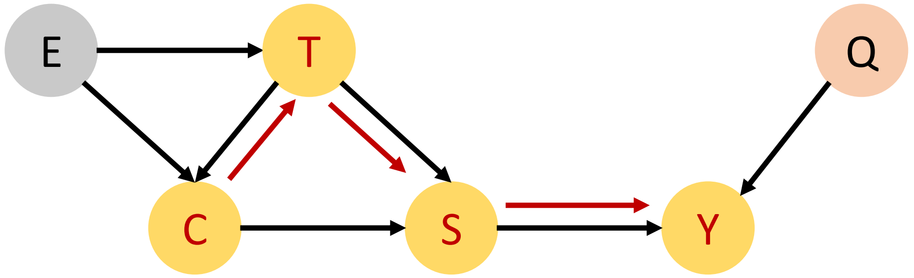


- Trigger is the confounder of context and predicted result.
- There exists a backdoor path $C \leftarrow T \rightarrow Y$

E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result

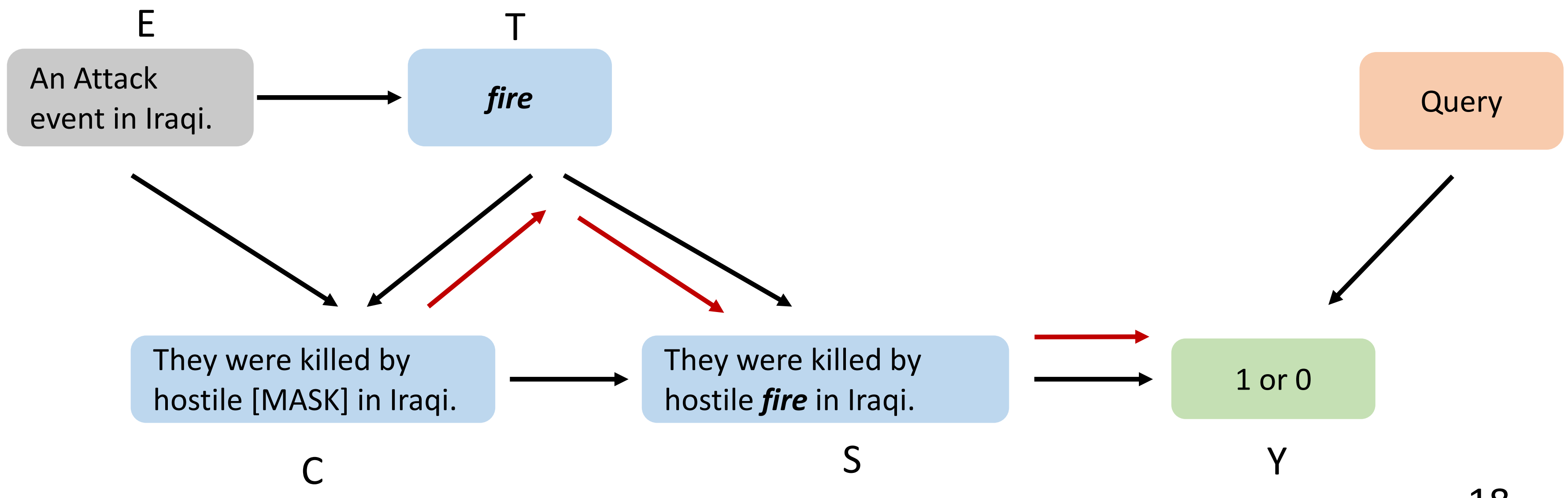


Structural Causal Model for FSED



- Trigger is the confounder of context and predicted result.
- There exists a backdoor path $C \leftarrow T \rightarrow Y$
- The backdoor path misleads the conventional learning procedure to mistakenly regard effects of triggers as the effects of contexts

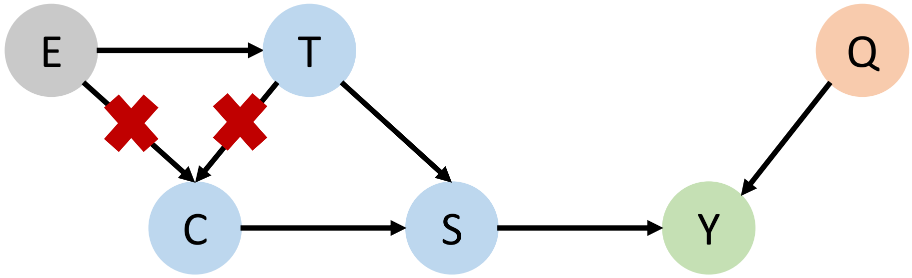
E : Event
T : Trigger set of support set
C : Context set of support set
S : Support set instances
Y : Predicted result



Outline

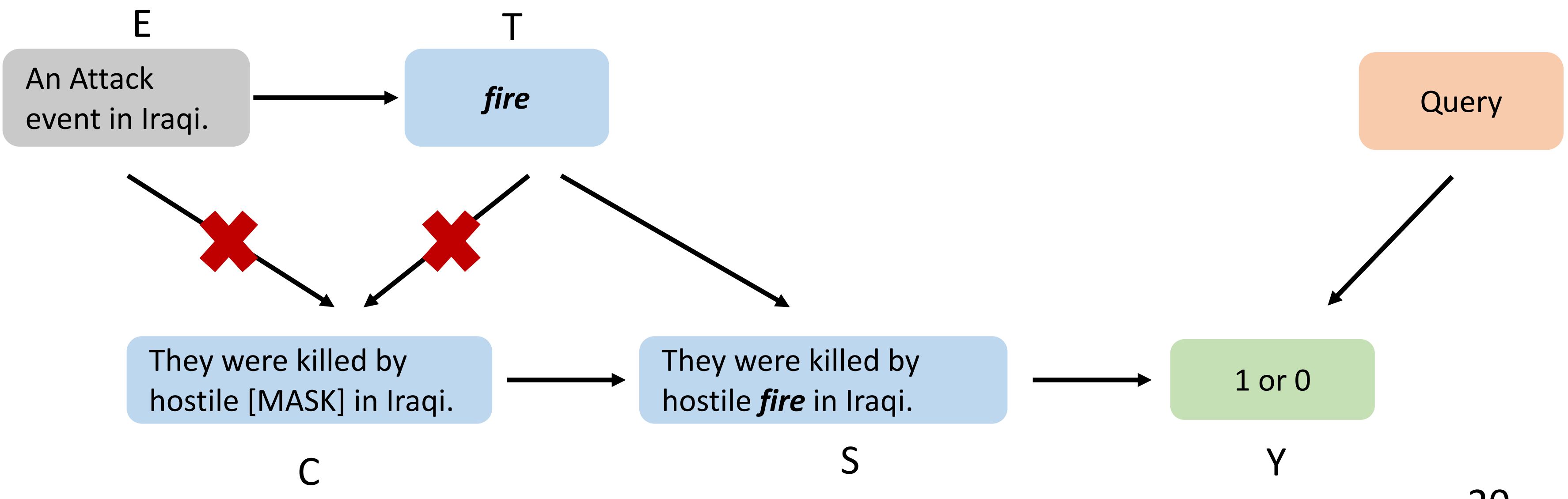
- Introduction
- Causal Intervention for Trigger Curse
- Experiments
- Conclusion

Causal Intervention for solving Trigger Curse

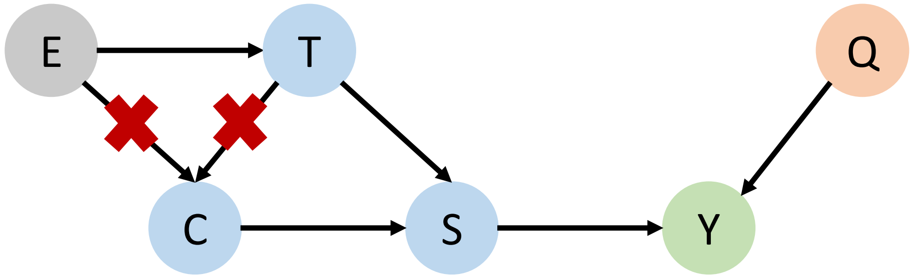


- Intervene on the context.

E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result



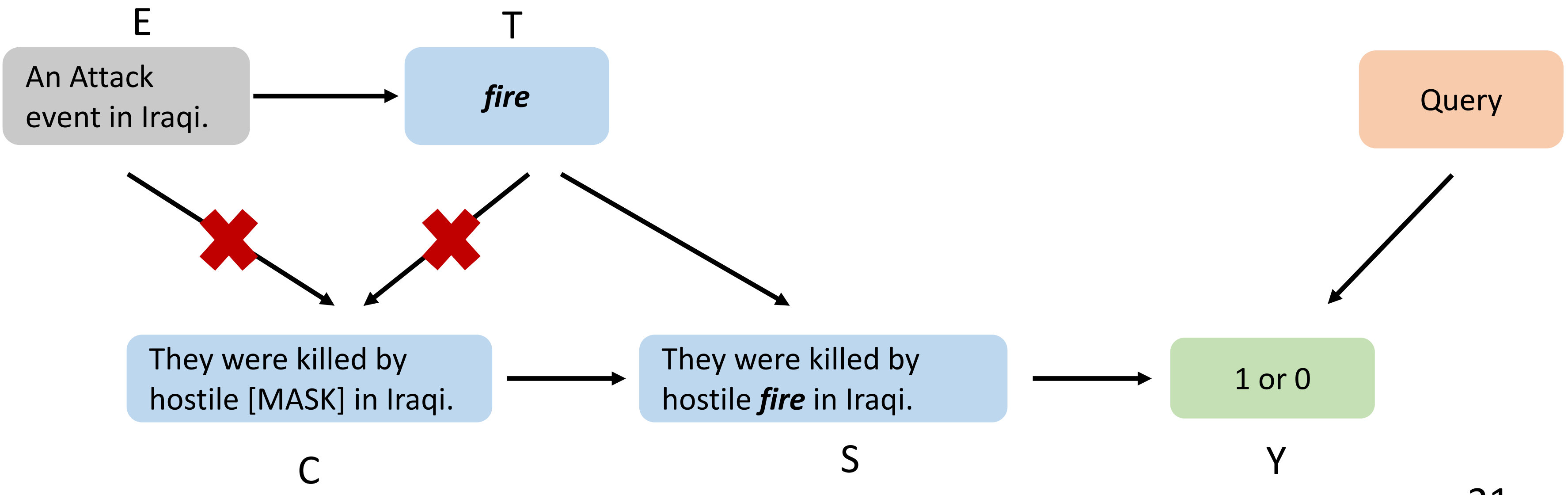
Causal Intervention for solving Trigger Curse



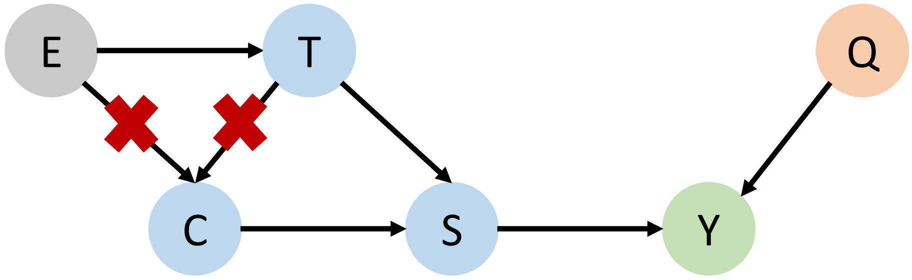
- Intervene on the context.
- Optimize the interventional distribution

$$P(Y|do(C = \mathrm{C}), E = e, Q = q)$$

E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result



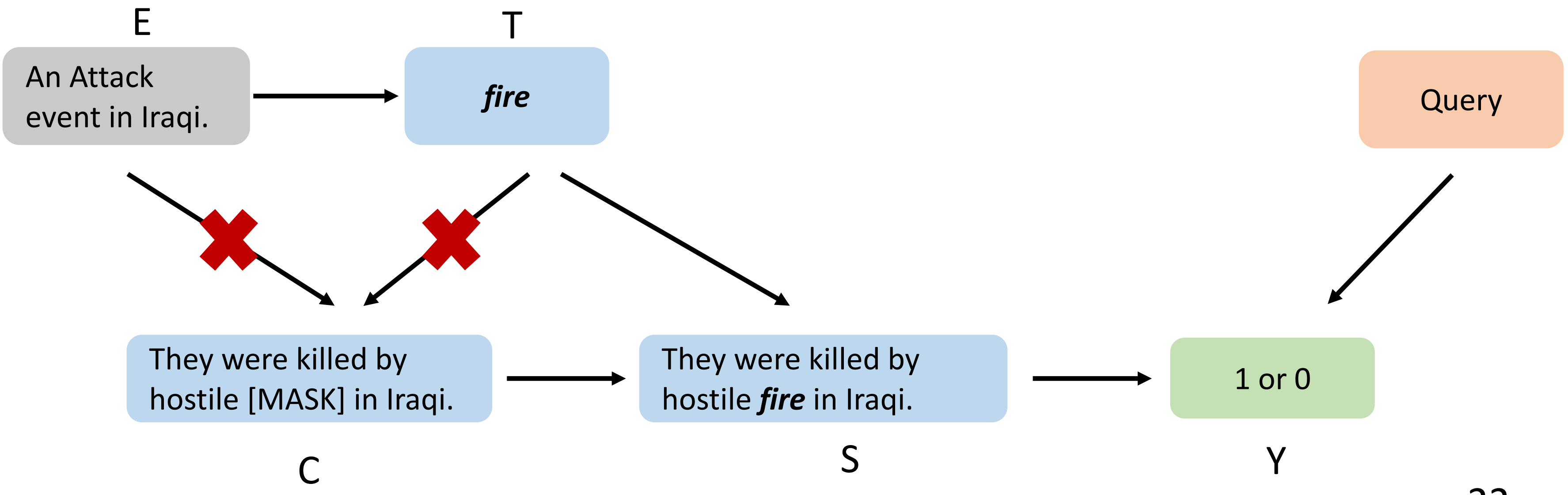
Causal Intervention for solving Trigger Curse



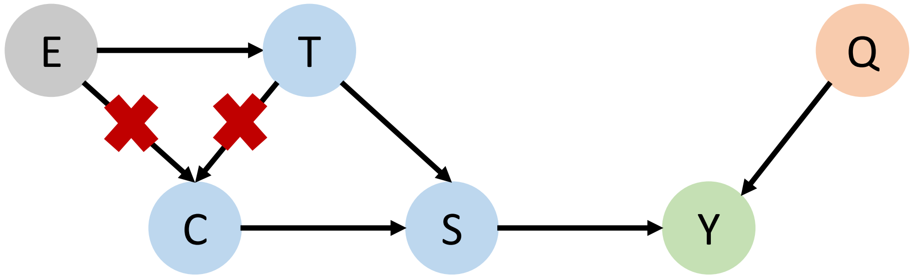
- Intervene on the context.
- Optimize the interventional distribution

$$P(Y|do(C=\mathrm{C}), E=e, Q=q)$$
$$=\sum_{s\in S}\sum_{t\in T} P(Y|s,q)P(s|\mathrm{C},t)P(t|e)$$

E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result



Causal Intervention for solving Trigger Curse

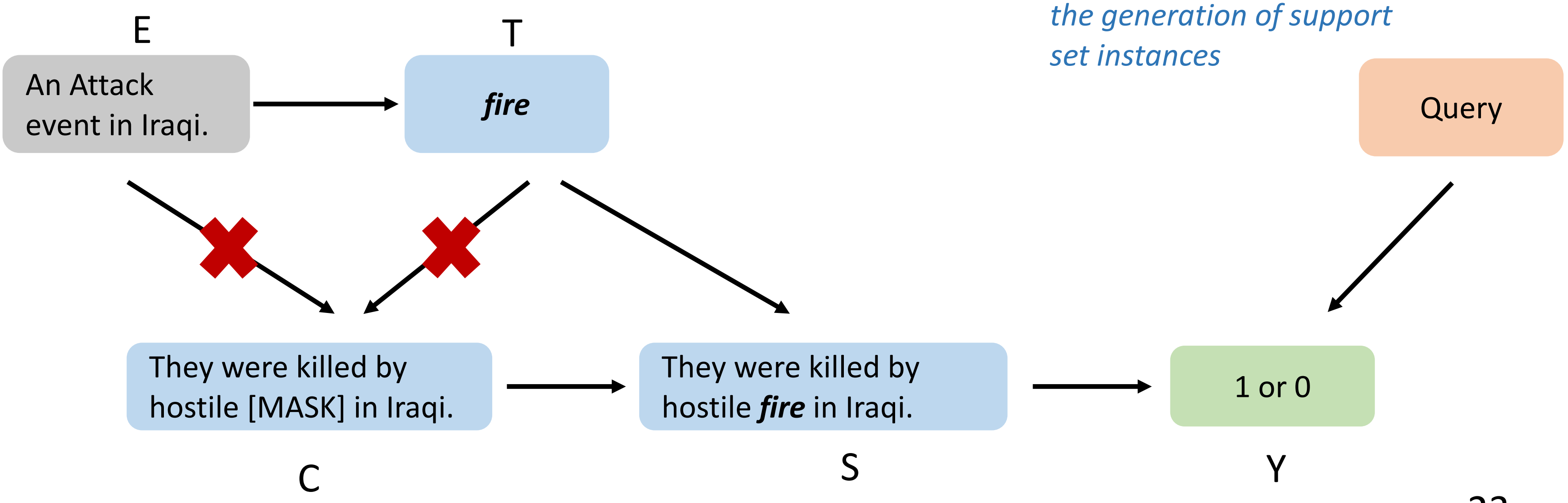


- Intervene on the context.
- Optimize the interventional distribution

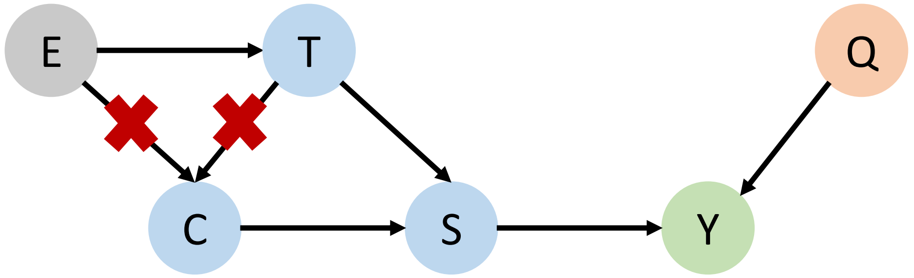
$$P(Y|do(C=\mathrm{C}),E=e,Q=q)$$
$$=\sum_{s\in S}\sum_{t\in T}P(Y|s,q)\boxed{P(s|\mathrm{C},t)}P(t|e)$$

the generation of support set instances

E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result



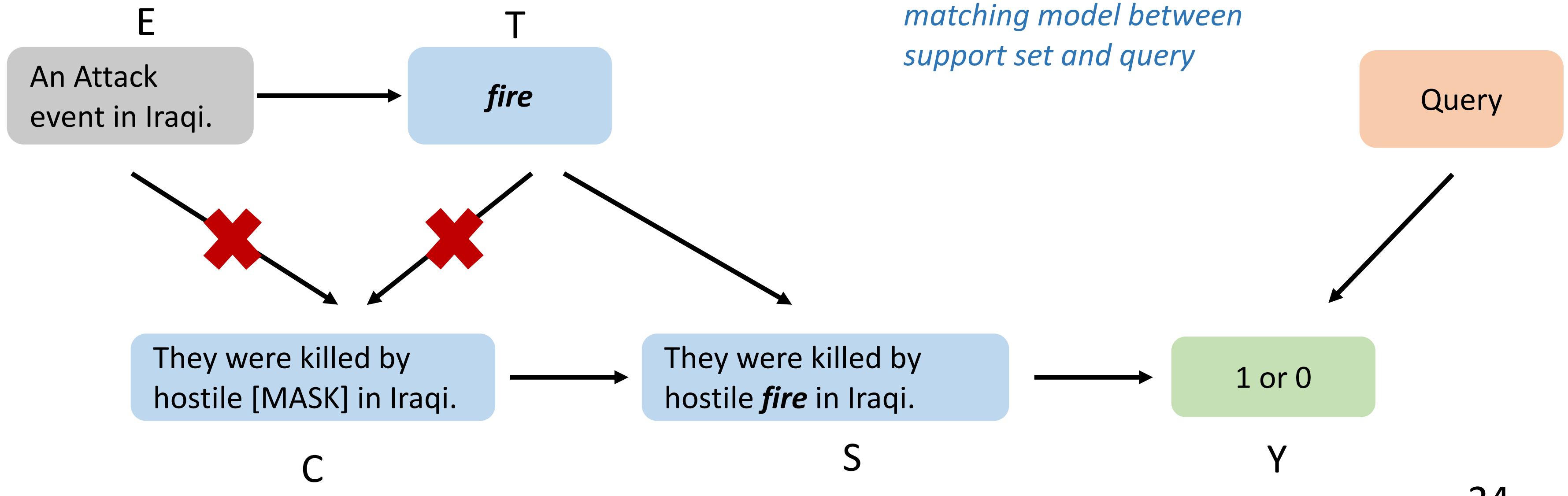
Causal Intervention for solving Trigger Curse



- Intervene on the context.
- Optimize the interventional distribution

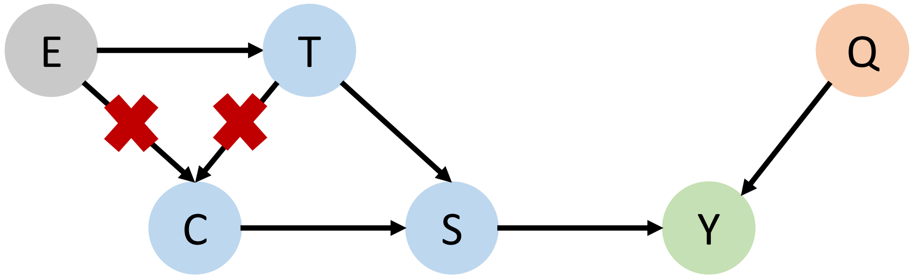
$$P(Y|do(C=\mathbb{C}),E=e,Q=q)$$
$$=\sum_{s\in S}\sum_{t\in T}\boxed{P(Y|s,q)}P(s|\mathbb{C},t)P(t|e)$$

matching model between support set and query



E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result

Causal Intervention for solving Trigger Curse

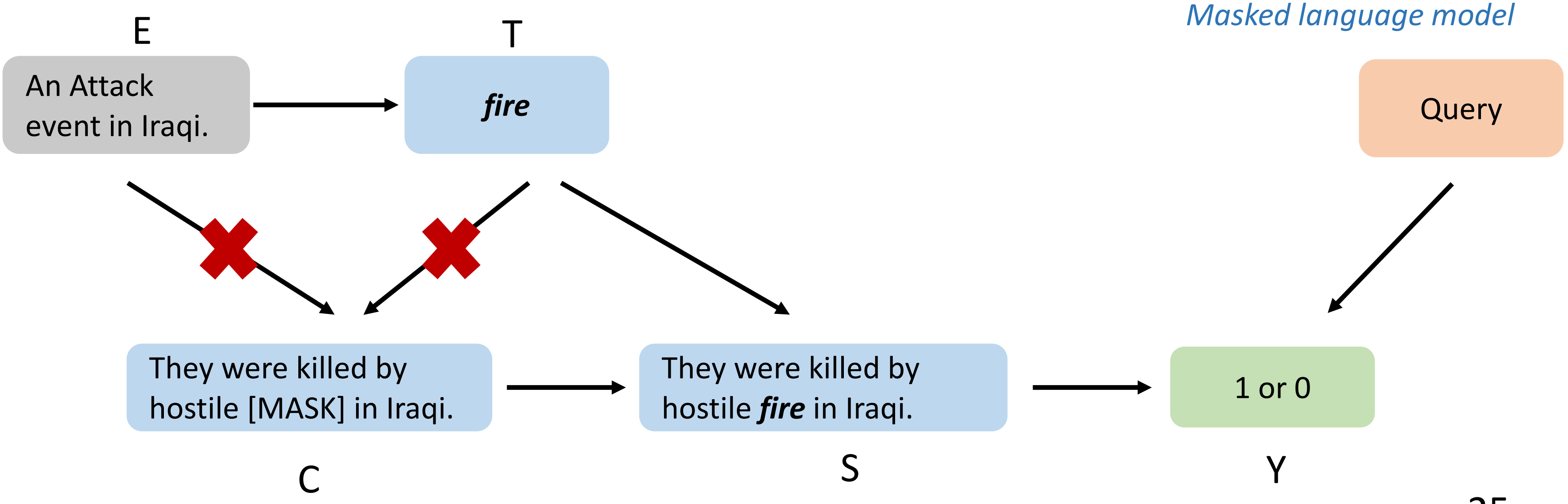


- Intervene on the context.
- Optimize the interventional distribution

$$P(Y|do(C = \mathrm{C}), E = e, Q = q)$$
$$= \sum_{s \in S} \sum_{t \in T} P(Y|s, q) P(s|\mathrm{C}, t) \boxed{P(t|e)}$$

Masked language model

E : event
T : Trigger set of support set
C : Context set of support set
S : support set instances
Y : Predicted result



Outline

- Introduction
- Causal Intervention for Trigger Curse
- Experiments
- Conclusion

Experiment Settings

- ACE05 dataset
  - Training set: 20 event types
  - Dev set: 10 event types
  - Test set: 10 event type
- MAVEN dataset
  - Training set: 120 event types
  - Dev set: 45 event types
  - Test set: 45 event types
- KBP17 dataset
  - Training set: 25 event types
  - Training set: 13 event types
  - Training set: 13 event types

Experiment Settings

- ACE05 dataset
  - Training set: 20 event types
  - Dev set: 10 event types
  - Test set: 10 event type
- MAVEN dataset
  - Training set: 120 event types
  - Dev set: 45 event types
  - Test set: 45 event types
- KBP17 dataset
  - Training set: 25 event types
  - Training set: 13 event types
  - Training set: 13 event types

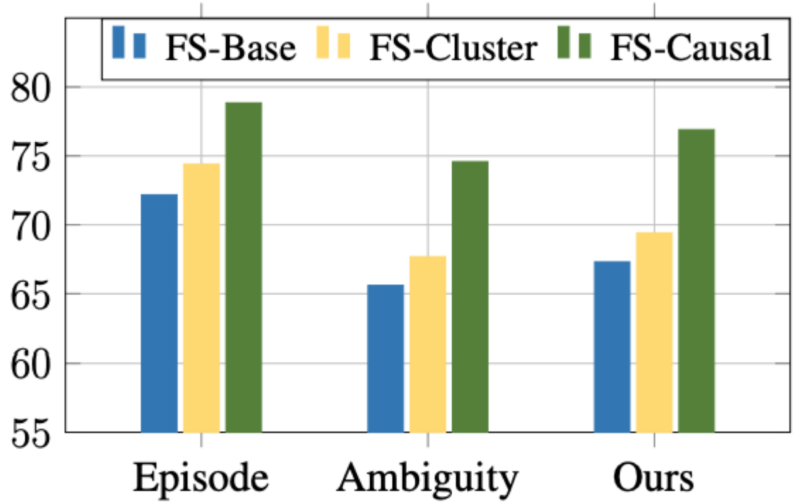
- 5 shot setting
  - Support set: 5 instances for each event types
  - Query: the full test set

Experimental Results

	Model	ACE05		MAVEN		KBP17	
		Macro	Micro	Macro	Micro	Macro	Micro
Finetuing-based	Finetune	51.0±1.4	58.2±1.6	30.7±1.5	31.6±2.3	59.4±1.9	62.7±1.8
	Finetune*	39.9±1.1	45.5±0.7	20.8±1.0	20.6±0.8	45.0±0.7	47.3±0.6
	Pretrain+Finetune	22.9±6.0	20.3±4.3	20.9±4.6	16.9±5.2	35.1±5.9	30.1±5.5
	Pretrain+Finetune*	14.6±3.3	15.6±3.4	12.5±3.8	14.9±4.0	23.4±6.8	25.8±6.3
Prototypical Net	FS-Base	63.8±2.8	67.3±2.7	44.7±1.4	44.5±2.0	65.5±2.7	67.3±3.1
	FS-LexFree	52.7±2.9	53.9±3.2	25.6±1.0	21.8±1.4	60.7±2.5	61.4±2.8
	FS-ClusterLoss	64.9±1.5	69.4±2.0	44.2±1.2	44.0±1.2	65.5±2.3	67.1±2.4
	FS-Causal (Ours)	**73.0**±2.2	**76.9**±1.4	**52.1**±0.2	**55.0**±0.4	**70.9**±0.6	**73.2**±0.9
Relation Net	FS-Base	65.7±3.7	68.7±4.5	52.4±1.4	56.0±1.4	**67.2**±1.5	71.2±1.4
	FS-LexFree	59.3±3.5	60.1±3.9	43.8±1.9	45.9±2.4	61.9±2.4	65.4±2.8
	FS-ClusterLoss	57.6±2.3	60.2±3.2	46.3±1.1	51.8±1.4	56.8±3.0	62.1±2.5
	FS-Causal (Ours)	**67.2**±1.4	**71.8**±1.9	**53.0**±0.5	**57.0**±0.9	66.4±0.4	**72.0**±0.6

- Our method based on metric-based method
- Our method can surpasses all the baselines .

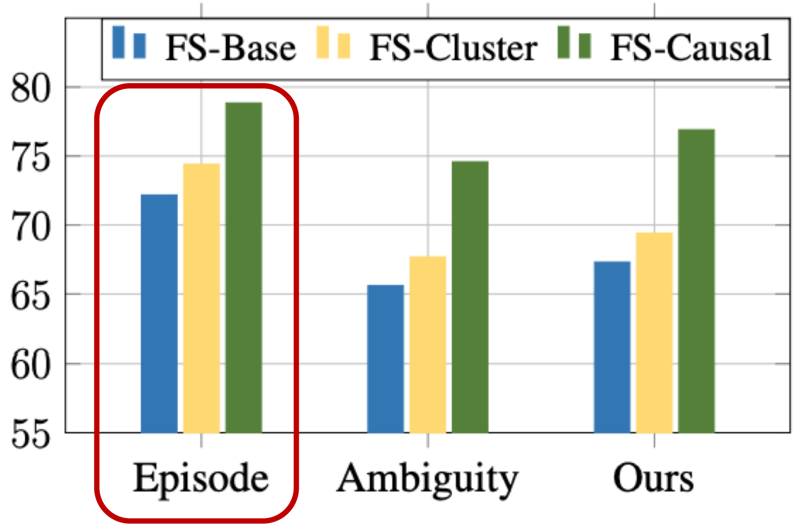
Different Settings



Experiment setting

- Episode: Conventional 5-way 5-shot setting
- Ambiguity: Adding some negative ambiguity event instances are sampled to the query.

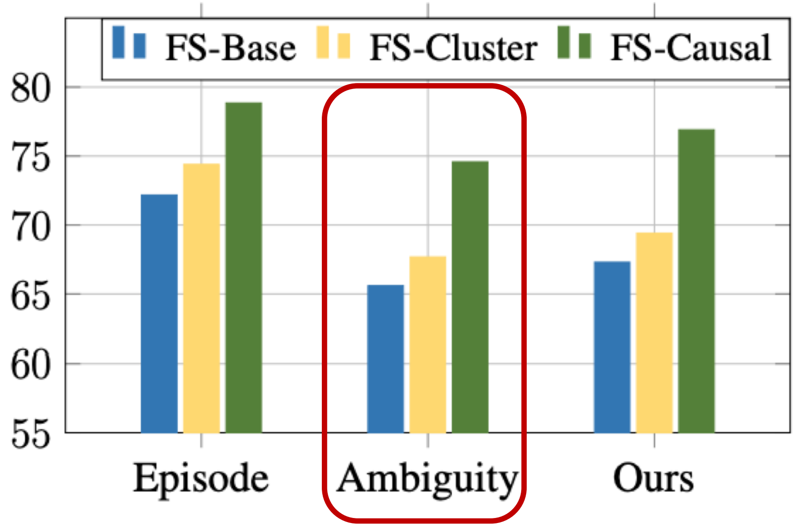
Different Settings



Experiment setting

- All models can achieve better performance on Episode setting.
    - Correctly recognize high-frequent triggers can achieve good performance in this setting.

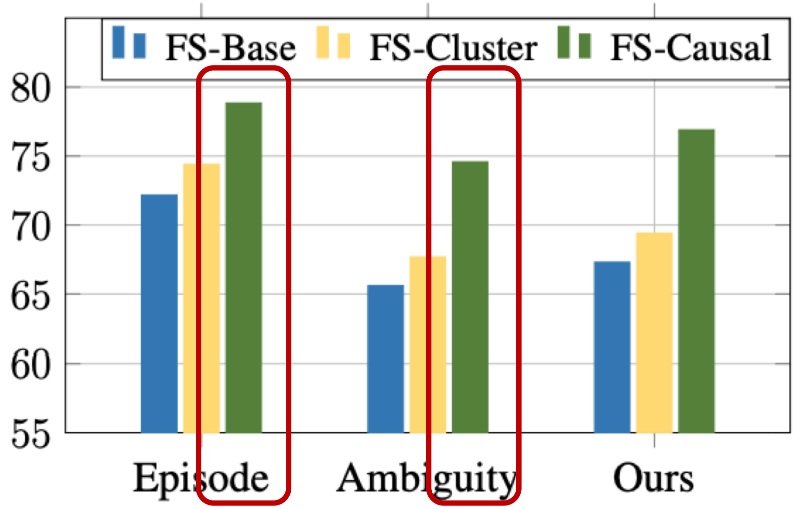
Different Settings



Experiment setting

- The performance of all models dropped on Ambiguity setting.
  - Trigger overfitting has a significant impact on FSED

Different Settings



Experiment setting

- Our method still maintains good performance on Ambiguity.
  - our method can alleviate the trigger curse problem by optimizing towards the underlying causality.

Conclusion

- Trigger Curse in Few-shot Event Detection
- Structural causal model for Few-shot Event Detection
- Causal intervention via backdoor adjustment

Thanks!
Any Question?

Source Code: https://github.com/chen700564/causalFSED