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, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
{jiawei2020}@iscas.ac.cn
{hongyu,xianpei,sunle}@iscas.ac.cn

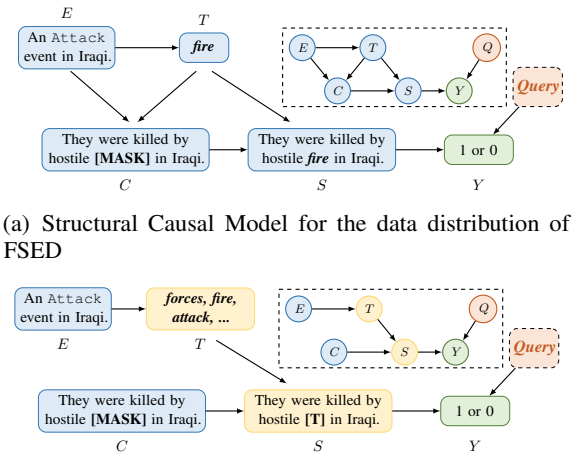
Abstract

Event detection has long been troubled by the *trigger curse*: overfitting the trigger will harm the generalization ability while underfitting it will hurt the detection performance. This problem is even more severe in few-shot scenario. In this paper, we identify and solve the trigger curse problem in few-shot event detection (FSED) from a causal view. By formulating FSED with a structural causal model (SCM), we found that the trigger is a confounder of the context and the result, which makes previous FSED methods much easier to overfit triggers. To resolve this problem, we propose to intervene on the context via backdoor adjustment during training. Experiments show that our method significantly improves the FSED on ACE05, MAVEN and KBP17 datasets.

1 Introduction

Event detection (ED) aims to identify and classify event triggers in a sentence, e.g., detecting an `Attack` event triggered by *fire* in "They killed by hostile fire in Iraqi". Recently, supervised ED approaches have achieved promising performance (Chen et al., 2015; Nguyen and Grishman, 2015; Nguyen et al., 2016; Lin et al., 2018, 2019b,a; Du and Cardie, 2020; Liu et al., 2020a; Lu et al., 2021), but when adapting to new event types and domains, a large number of manually annotated event data is required which is expensive. By contrast, few-shot event detection (FSED) aims to build effective event detectors that are able to detect new events from instances (query) with a few labeled instances (support set). Due to their ability to classify novel types, many few-shot algorithms have been used in FSED, e.g., metric-based methods like Prototypical Network (Lai et al., 2020; Deng et al., 2020; Cong et al., 2021).

Unfortunately, there has long been a "trigger curse" which troubles the learning of event detection models, especially in few-shot scenario (Bronstein et al., 2015; Liu et al., 2017; Chen et al., 2018; Liu et al., 2019; Ji et al., 2019). For many event types, their triggers are dominated by several popular words, e.g., the `Attack` event type is dominated by *war, attack, fight, fire, bomb* in ACE05. And we found the top 5 triggers of each event type cover 78% of event occurrences in ACE05. Due to the trigger curse, event detection models nearly degenerate to a trigger matcher, ignore the majority of contextual information and mainly rely on whether the candidate word matches the dominant triggers. This problem is more severe in FSED: since the given support instances are very sparse and lack diversity, it is much easier to overfit the trigger of the support instances. An intuitive solution for the trigger curse is to erase the trigger information in instances and forces the model to focus more on the context. Unfortunately, due to the decisive role of triggers, directly wiping out the trigger information commonly hurts the performance (Lu et al., 2019; Liu et al., 2020b). Some previous approaches try to tackle this problem by introducing more di-



(a) Structural Causal Model for the data distribution of FSED

(b) The data distribution of FSED after causal intervention

Figure 1: Illustration of the causal intervention strategy proposed in this paper. The graph includes the event E, the trigger set T, the context set C, the support instance S, the prediction Y and the query instance Q.

*Corresponding authors.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8078–8088
November 7–11, 2021.

versified context information like event argument information (Liu et al., 2017, 2019; Ji et al., 2019) and document-level information (Ji and Grishman, 2008; Liao and Grishman, 2010; Duan et al., 2017; Chen et al., 2018). However, rich context information is commonly not available for FSED, and therefore these methods can not be directly applied.

In this paper, we revisit the trigger curse in FSED from a causal view. Specifically, we formulate the data distribution of FSED using a trigger-centric structural causal model (SCM) (Pearl et al., 2016) shown in Figure 1(a). Such trigger-centric formulation is based on the fact that, given the event type, contexts have a much lower impact on triggers, compared with the impact of triggers on contexts. This results in the decisive role of triggers in event extraction, and therefore conventional event extraction approaches commonly follow the trigger-centric procedure (i.e., identifying triggers first and then using triggers as an indicator to find arguments in contexts). Furthermore, the case grammar theory in linguistics (Fillmore, 1967) also formulate the language using such trigger/predicate-centric assumption, and have been widely exploited in many NLP tasks like semantic role labeling (Gildea and Jurafsky, 2002) and abstract meaning representation (Banarescu et al., 2013).

From the SCM, we found that T(trigger set) is a confounder of the C(context set) and the Y(result), and therefore there exists a backdoor path $C \leftarrow T \rightarrow Y$. The backdoor path explains why previous FSED models disregard contextual information: it misleads the conventional learning procedure to mistakenly regard effects of triggers as the effects of contexts. Consequently, the learning criteria of conventional FSED methods are optimized towards spurious correlation, rather than capturing causality between C and Y. To address this issue, we propose to intervene on context to block the information from trigger to context. Specifically, we apply backdoor adjustment to estimate the interventional distribution that is used for optimizing causality. Furthermore, because backdoor adjustment relies on the unknown prior confounder (trigger) distribution, we also propose to estimate it based on contextualized word prediction.

We conducted experiments on ACE05[1], MAVEN[2] and KBP17[3] datasets. Experiments show that causal intervention can significantly alleviate trigger curse, and therefore the proposed method significantly outperforms previous FSED methods.

2 Structural Causal Model for FSED

This section describes the structural causal model (SCM) for FSED, illustrated in Figure 1(a). Note that, we omit the causal structure of the query for simplicity since it is the same as the support set. Concretely, the SCM formulates the data distribution of FSED: 1) Starting from an event E we want to describe (in Figure 1(a) is an `Attack` in Iraqi). 2) The path $E \rightarrow T$ indicates the trigger decision process, i.e., selecting words or phrases (in Figure 1(a) is *fire*) which can almost clearly express the event occurrence (Doddington et al., 2004). 3) The path $E \rightarrow C \leftarrow T$ indicates that a set of contexts are generated depending on both the event and the trigger, which provides background information and organizes this information depending on the trigger. For instance, the context "They killed by hostile [*fire*] in Iraqi" provides the place, the role and the consequences of the event, and this information is organized following the structure determined by *fire*. 4) an event instance is generated by combining one of the contexts in C and one of the triggers in T via the path $C \rightarrow S \leftarrow T$. 5) Finally, a matching between query and support set is generated through $S \rightarrow Y \leftarrow Q$.

Conventional learning criteria for FSED directly optimize towards the conditional distribution $P(Y|S,Q)$. However, from the SCM, we found that the backdoor path $C \leftarrow T \rightarrow Y$ pass on associations (Pearl et al., 2016) and mislead the learning with spurious correlation. Consequently, the learning procedure towards $P(Y|S,Q)$ will mistakenly regard the effects of triggers as the effects of contexts, and therefore overfit the trigger information.

3 Causal Intervention for Trigger Curse

Based on the SCM, this section describes how to resolve the trigger curse via causal intervention.

Context Intervention. To block the backdoor path, we intervene on the context C and the new context-intervened SCM is shown in Figure 1(b). Given support set s, event set e of s, context set $\mathcal{C}$ of s and query instance q, we optimize the interventional distribution $P(Y|do(C=\mathcal{C}), E=e, Q=q)$ rather than $P(Y|S=s, Q=q)$, where $do(\cdot)$ denotes causal intervention operation. By interven-

[1] https://catalog.ldc.upenn.edu/LDC2006T06
[2] https://github.com/THU-KEG/MAVEN-dataset
[3] https://tac.nist.gov/2017/KBP/data.html

ing, the learning objective of models changes from optimizing correlation to optimizing causality.

Backdoor Adjustment. Backdoor adjustment is used to estimate the interventional distribution[4]:

$$P(Y|do(C{=}\mathcal{C}), E{=}e, Q{=}q) = \sum_{t\in T}\sum_{s\in S} P(Y|s,q)P(s|\mathcal{C},t)P(t|e), \quad (1)$$

where $P(s|\mathcal{C},t)$ denotes the generation of s from the trigger and contexts. $P(s|\mathcal{C},t) = 1/|\mathcal{C}|$ if and only if the context of s in $\mathcal{C}$ and the trigger of s is t. $P(Y|s,q) \propto \phi(s,q;\theta)$ is the matching model between q and s parametrized by θ.

Estimating $P(t|e)$ via Contextualized Prediction. The confounder distribution $P(t|e)$ is unknown because E is a hidden variable. Since the event argument information is contained in $\mathcal{C}$, we argue that $P(t|e) \propto M(t|\mathcal{C})$ where $M(\cdot|\mathcal{C})$ indicates a masked token prediction task (Taylor, 1953) which is constructed by masking triggers in the support set. In this paper, we use masked language model to calculate $P(t|e)$ by first generating a set of candidate triggers through the context: $T_c = \{t_i|i = 1, 2, \ldots\} \cup \{t_0\}$, where t_i is the i-th predicted token and t_0 is the original trigger of the support set instance, then $P(t|e)$ is estimated by averaging logit obtained from the MLM:

$$P(t_i|e) = \begin{cases} \lambda & i = 0 \\ (1-\lambda)\dfrac{\exp(l_i)}{\sum_j \exp(l_j)} & i \neq 0 \end{cases} \quad (2)$$

where l_i is the logit for the i^{th} token. To reduce the noise introduced by MLM, we assign an additional hyperparameter $\lambda \in (0, 1)$ to t_0.

Optimizing via Representation Learning. Given the interventional distribution, FSED model can be learned by minimizing the loss function on it:

$$\begin{aligned}\mathcal{L}(\theta) &= -\sum_{q\in Q} f(P(Y|do(\mathcal{C}), e, q; \theta)) \\ &= -\sum_{q\in Q} f(\sum_{t\in T}\sum_{s\in S} P(Y|s,q;\theta)P(s|\mathcal{C},t)P(t|e))\end{aligned} \quad (3)$$

where Q is training queries and f is a strict monotonically increasing function. However, the optimization of $\mathcal{L}(\theta)$ needs to calculate every $P(Y|s,q;\theta)$, which is quite time-consuming. To this end, we propose a surrogate learning criteria $\mathcal{L}_{SG}(\theta)$ to optimize the causal relation based on representation learning:

$$\mathcal{L}_{SG}(\theta) = -\sum_{q\in Q} g(\mathbf{R}(q;\theta), \sum_{t\in T}\sum_{s\in S} P(s|\mathcal{C},t)P(t|e)\mathbf{R}(s;\theta)) \quad (4)$$

Here $\mathbf{R}$ is a representation model which inputs s or q and outputs a dense representation. $g(\cdot,\cdot)$ is a distance metric measuring the similarity between two representations. Such loss function is widely used in many metric-based methods (e.g., Prototypical Networks and Relation Networks). In the Appendix, we prove $\mathcal{L}_{SG}(\theta)$ is equivalent to $\mathcal{L}(\theta)$.

4 Experiments

4.1 Experimental Settings

Datasets.[5] We conducted experiments on ACE05, MAVEN (Wang et al., 2020c) and KBP17 datasets. We split train/dev/test sets according to event types and we use event types with more instances for training, the other for dev/test. To conduct 5-shot experiments, we filter event types less than 6 instances. Finally, for ACE05, its train/dev/test set contains 3598/140/149 instances and 20/10/10 types respectively, for MAVEN, those are 34651/1494/1505 instances and 120/45/45 types, for KBP17, those are 15785/768/792 instances and 25/13/13 types.

Task Settings. Different from episode evaluation in Lai et al. (2020) and Cong et al. (2021), we employ a more practical event detection setting inspired by Yang and Katiyar (2020) in Few-shot NER. We randomly sample few instances as support set and all other instances in the test set are used as queries. A support set corresponds to an event type and all types will be evaluated by traversing each event type. Models need to detection the span and type of triggers in a sentence. We also compared the results across settings in Section 4.3. We evaluate all methods using macro-F1 and micro-F1 scores, and micro-F1 is taken as the primary measure.

Baselines. We conduct experiments on two metric-based methods: Prototypical Network (Snell et al., 2017) and Relation Network (Sung et al., 2018), which are referred as **FS-Base**. Based on these models, we compare our causal intervention method (**FS-Casual**) with 1) **FS-LexFree** (Lu et al., 2019), which address overfit triggers via adversarial learning, we use their lexical-free encoder;

[4]The proof is shown in Appendix

[5]Our source codes are openly available at https://github.com/chen700564/causalFSED

	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

Table 1: F1 score of 5-shot FSED on test set. * means fixing the parameters of encoder when finetuning. ± is the standard deviation of 5 random training rounds.

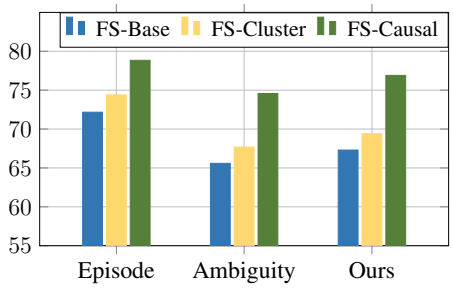


Figure 2: Micro F1 of prototypical network with different settings on ACE05 test set.

2) **FS-ClusterLoss** (Lai et al., 2020), which add two auxiliary loss functions when training. Furthermore, we compare our method with models finetuned with support set (**Finetune**) and pretrained using the training set (**Pretrain**). $\text{BERT}_{\text{base}}$ (uncased) is used as the encoder for all models and MLM for trigger collection.

4.2 Experimental Results

The performance of our method and all baselines is shown in Table 1. We can see that:

1) **By intervening on the context in SCM and using backdoor adjustment during training, our method can effectively learn FSED models.** Compared with the original metric-based models, our method achieves 8.7% and 1.6% micro-F1 (average) improvement in prototypical network and relation network respectively.

2) **The causal theory is a promising technique for resolving the trigger cruse problem.** Notice that FS-LexFree cannot achieve the competitive performance with the original FS models, which indicates that trigger information is import and underfitting triggers will hurt the detection performance. This verifies that trigger curse is very challenging and causal intervention can effectively resolve it.

3) **Our method can achieve state-of-the-art FSED performance.** Compared with best score in baselines, our method gains 7.5%, 1.0%, and 2.0% micro-F1 improvements on ACE05, MAVEN and KBP17 datasets respectively.

4.3 Effect on Different Settings

To further demonstrate the effectiveness of the proposed method, we also conduct experiments under different FSED settings: 1) The primal episode-based settings (**Episode**), which is the 5+1-way 5-shot settings in Lai et al. (2020). 2) Episode + ambiguous instances (**Ambiguity**), which samples some additional negative query instances that include words same as triggers in support set to verify whether models overfit the triggers.

The performance of different models with different settings is shown in Figure 2. We can see that: 1) Generally speaking, all models can achieve better performance on **Episode** because correctly recognize high-frequent triggers can achieve good performance in this setting. Consequently, the performance under this setting can not well represent how FSED is influenced by trigger overfitting. 2) The performance of all models dropped on **Ambiguity** setting, which suggests that trigger overfitting has a significant impact on FSED. 3) Our method still maintains good performance on **Ambiguity**, which indicates that our method can alleviate the trigger curse problem by optimizing towards the underlying causality.

4.4 Case Study

We select ambiguous cases (in Table 2) to better illustrate the effectiveness of our method. For *Query 1*, FS-Base wrongly detects the word *run* to be a trigger word. In *Support set 1*, *run* means nomi-

Support set 1	I mean , I 'd like to - – I 'd like to see the Greens *[Nominate]run[/Nominate]* David Cobb again.
Query 1	Release a known terrorist to run the PLO and that will bring about peace
FS-Base	Release a known terrorist to *[Nominate]run[/Nominate]* the PLO and that will bring about peace
FS-Causal	Release a known terrorist to run the PLO and that will bring about peace
Support set 2	They were *[Suspicion]suspected[/Suspicion]* of having facilitated the suicide bomber.
Query 2	A fourth suspect, Osman Hussein, was arrested in Rome, Italy, and later extradited to the UK.
FS-Base	A fourth *[Suspicion]suspect[/Suspicion]*, Osman Hussein, was arrested in Rome, Italy, and later extradited to the UK.
FS-Causal	A fourth suspect, Osman Hussein, was arrested in Rome, Italy, and later extradited to the UK.

Table 2: Ambiguous cases from ACE05 and MAVEN test set. The results are based on prototypical network and *Support set* means one instance in the support set.

nating while *run* means managing in *Query 1*. FS-Base fails to recognize such different sense of word under context. For *Query 2*, FS-Base makes mistake again on the ambiguous word *suspect*. Even though *suspect* is the noun form of *suspected* in *Support set 2*, it does not trigger a `Suspicion` event in *Query 2*. In contract to FS-Base, our approach is able to handle both cases correctly, illustrating its effectiveness.

5 Related Work

Causal Inference. Causal inference aims to make reliable predictions using the causal effect between variables (Pearl, 2009). Many studies have used causal theory to improve model robustness (Wang et al., 2020a,b; Qi et al., 2020; Tang et al., 2020b; Zeng et al., 2020). Recently, backdoor adjustment has been used to remove the spurious association brought by the confounder (Tang et al., 2020a; Zhang et al., 2020; Yue et al., 2020; Liu et al., 2021; Zhang et al., 2021).

Few-shot Event Detection. Few-shot event detection has been studied in many different settings. Bronstein et al. (2015) collect some seed triggers, then detect unseen event with feature-based method. Deng et al. (2020) decompose FSED into two subtasks: trigger identification and few-shot classification. Feng et al. (2020) adopt a sentence-level few-shot classification without triggers. Lai et al. (2020) and Cong et al. (2021) adopt N+1-way few-shot setting that is closest to our setting.

6 Conclusions

This paper proposes to revisit the trigger curse in FSED from a causal view. Specifically, we identify the cause of the trigger curse problem from a structural causal model, and then solve the problem through casual intervention via backdoor adjustment. Experimental results demonstrate the effectiveness and robustness of our methods.

Acknowledgments

We thank the reviewers for their insightful comments and helpful suggestions. This research work is supported by National Key R&D Program of China under Grant 2018YFB1005100, the National Natural Science Foundation of China under Grants no. 62106251 and 62076233, and in part by the Youth Innovation Promotion Association CAS(2018141).

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 372–376, Beijing, China. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40, Online. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 352–361.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction. *CoRR*, abs/2010.11325.

Charles Fillmore. 1967. The case for case.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Yuze Ji, Youfang Lin, Jianwei Gao, and Huaiyu Wan. 2019. Exploiting the entity type sequence to benefit event detection. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 613–623, Hong Kong, China. Association for Computational Linguistics.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Nugget proposal networks for Chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1565–1574, Melbourne, Australia. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019a. Cost-sensitive regularization for label confusion-aware event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5278–5283, Florence, Italy. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019b. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Fangchao Liu, Lingyong Yan, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Element intervention for open relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4683–4693, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Kang Liu. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6754–6761.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020a. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020b. How does context matter? on the robustness of event detection with context-selective mask generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2523–2532, Online. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376, Florence, Italy. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. 2020. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10860–10869.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020a. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020b. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020a. Visual commonsense r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10760–10770.

Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020b. Visual commonsense representation learning via causal inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 378–379.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020c. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. 2020. Interventional few-shot learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280, Online. Association for Computational Linguistics.

Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. 2020. Causal intervention for weakly-supervised semantic segmentation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021. De-biasing distantly supervised named entity recognition via causal intervention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4803–4813, Online. Association for Computational Linguistics.

A Proof of Backdoor Adjustment

We prove the backdoor adjustment for our SCM using the rules of do-calculus (Pearl, 1995).

For a causal graph G, let $G_{\overline{X}}$ denote the graph where all of the incoming edges to Node X are removed. let $G_{\underline{X}}$ denote the graph where all of the outgoing edges from Node X are removed. $\perp\!\!\!\perp_G$ denotes d-separation in G.

D-separation (Pearl, 2014): Two (sets of) nodes X and Y are d-separation by a set of nodes Z (i.e. $X \perp\!\!\!\perp_G Y|Z$) if all of the paths between (any node in) X and (any node in) Y are blocked by Z.

The rules of do-calculus are:

Rule 1

$$P(y|do(t), z, w) = P(y|do(t), w) \quad \text{if} \quad Y \perp\!\!\!\perp_{G_{\overline{T}}} Z|T, W$$

Rule 2

$$P(y|do(t), do(z), w) = P(y|do(t), z, w) \quad \text{if} \quad Y \perp\!\!\!\perp_{G_{\overline{T}\underline{Z}}} Z|T, W$$

Rule 3

$$P(y|do(t), do(z), w) = P(y|do(t), w) \quad \text{if} \quad Y \perp\!\!\!\perp_{G_{\overline{T}\underline{Z(W)}}} Z|T, W \tag{5}$$

where $Z(W)$ denotes the set of nodes of Z that aren't ancestors of any node of W in $G_{\overline{T}}$.

We can prove our interventional distribution $P(Y|do(C = \mathcal{C}), E = e)$:

Step 1 Using the law of total probability:

$$\begin{aligned} &P(Y|do(C = \mathcal{C}), E = e, Q = q) \\ =&\sum_{t\in T}\sum_{s\in S}[P(Y|do(\mathcal{C}), e, t, s, q)\times \\ &P(s, t|do(\mathcal{C}), e, q)] \end{aligned}$$

Step 2 Using the law of conditional probability:

$$\begin{aligned} &P(Y|do(C = \mathcal{C}), E = e, Q = q) \\ =&\sum_{t\in T}\sum_{s\in S}[P(Y|do(\mathcal{C}), e, t, s, q)\times \\ &P(s|do(\mathcal{C}), e, t, q)P(t|do(\mathcal{C}), e, q)] \end{aligned}$$

Step 3 Using the **Rule 3**:

$$\begin{aligned} &P(Y|do(C = \mathcal{C}), E = e, Q = q) \\ =&\sum_{t\in T}\sum_{s\in S}[P(Y|e, t, s, q)\times \\ &P(s|do(\mathcal{C}), e, t, q)P(t|e, q)] \end{aligned}$$

Step 4 Using the **Rule 1**:

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

Step 5 Using the **Rule 2**:

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

B Detailed Task Settings

One-way K-Shot Settings. We adopt One-way K-shot setting in our experiments, in which the support set in an episode contains one event type (called concerned event) and the query can contain any event type. The model aims to detect triggers of the concerned event in query and all types will be evaluated by traversing each event type. The support set and query in an episode can be formulated as follows:

$$\mathcal{S}=\{(S^1,E,Y^1),\ldots,(S^K,E,Y^K)\}$$

where $\mathcal{S}$ is the support set, E is the concerned event, $S^i=\{s_1^i,s_2^i,\ldots,s_{n_i}^i\}$ is the i-th sentence in support, s_j^i is the j-th token in S^i, $Y^i=\{y_1^i,y_2^i,\ldots,y_n^i\}$ is the labels of tokens in S_i and $y_j^i=1$ only if t_i is the trigger (or part of trigger) of concerned event, otherwise $y_j^i=0$.

$$\mathcal{Q}=\{Q^1,Q^2,\ldots,Q^M\}$$

where $\mathcal{Q}$ is the set of query and $Q^i=\{q_1^i,q_2^i,\ldots,q_{m_i}^i\}$ is the i-th query sentence and q_j^i is the j-th token in Q^i

The model is expected to output the concerned event in $\mathcal{Q}$:

$$\begin{aligned}\mathcal{O}_{\mathcal{Q}}=\{&(Q^1,E,T_1^1),\ldots,(Q^1,E,T_{n_1}^1),\\&(Q^2,E,T_1^2),\ldots,(Q^2,E,T_{n_2}^2),\\&\ldots,\\&(Q^M,E,T_1^M),\ldots,(Q^M,E,T_{n_M}^M)\}\end{aligned}$$

where $\mathcal{O}_{\mathcal{Q}}$ is the set of triggers of concerned event detected in $\mathcal{Q}$, T_k^i is the k-th trigger of concerned event in sentence Q^i and $n_i\geq 0$ means the number of triggers of concerned event in Q^i.

Evaluation We improve the traditional episode evaluation setting by evaluating the full test set. For each event type in test set, we randomly sample K instances as support set and all other instances are used as query. Following previous event detection works (Chen et al., 2015), the predicted trigger is correct if its event type and offsets match those of a gold trigger. We evaluate all methods using macro-F1 and micro-F1 scores, and micro-F1 is taken as the primary measure.

C Few-shot Event Detection Baselines

We use two metric-base methods in our experiments: Prototypical network (Snell et al., 2017) and Relation network (Sung et al., 2018), which contain an encoder component and a classifier component.

Encoder We use BERT (Devlin et al., 2019) to encoder the support set and the query. Given a sentece $\mathbf{X}=\{x_1,x_2,\ldots,x_n\}$, BERT encodes the sequence and output the represent of each token in $\mathbf{X}$: $\mathbf{R}=\{\mathbf{r_1},\mathbf{r_2},\ldots,\mathbf{r_n}\}$. After obtaining the feature representation of the support set, we calculate the prototype of the categories (concerned event and other):

$$\mathbf{p}_i=\frac{1}{|\mathcal{R}_i|}\sum_{\mathbf{r}\in\mathcal{R}_i}\mathbf{r},\quad i=0,1$$

where $\mathbf{p}_i$ is the prototype of category i, $\mathcal{R}_i$ is the set of feature representation of tokens that labeled with $y=i$ in support set.

Classifier The models classify each token in query based on its similarity to the prototype.

We first calculate the similarity between prototype and token in query.

$$s_{i,j,k}=g(\mathbf{p}_k,\mathbf{q}_j^i),\quad k=0,1 \tag{6}$$

where $g(\mathbf{x},\mathbf{y})$ measures the similarity between $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{q}_j^i$ is the represent of j-th token in i-th query sentence.

Then we calculate the probability distribution of token q_j^i:

$$P(Y|q_j^i,\mathcal{S})=\text{Softmax}(s_{i,j,0},s_{i,j,1}) \tag{7}$$

During training, we use the Cross-Entropy loss on each token of query. And the support set and the query are randomly sampled from the training set.

When evaluating, we treat the labels as IO tagging schemes, and adjacent I are considered to be the same trigger so that we can handle a trigger with multiple tokens.

Similarity Functions For prototypical network, the similarity in Equation 6 is Euclidean distance. For relation network, we calculate similarity using neural networks. Unlike the original paper, we find the following calculation to be more efficient: $g(\mathbf{p}_k, \mathbf{q}_i^j) = F(\mathbf{p}_k \oplus \mathbf{q}_i^j \oplus |\mathbf{p}_k - \mathbf{q}_i^j|)$ where $\oplus$ means concatenation vectors and F is two-layer feed-forward neural networks with a ReLU function on the first layer.

D Proof of Loss Function

We prove $\mathcal{L}_{SG}(\theta)$ is equivalent to $\mathcal{L}(\theta)$, which indicates that minimizing $\mathcal{L}_{SG}(\theta)$ is equivalent to minimizing $\mathcal{L}(\theta)$. At first , we define a function $\phi(s, q) \propto P(Y|s, q; \theta)$ and then we need to prove that $g(\sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t)\mathbf{r}_s, \mathbf{q}) = f(\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\phi(s, q))$.

From Appendix-A, we can obtain:

$$\begin{aligned}
&\sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t) \\
=&\sum_{t\in T}\sum_{s\in S} P(s, t|do(\mathcal{C}), e, q) \\
=&1
\end{aligned}$$

D.1 Prototypical Network

For prototypical network, $g(\mathbf{r}, \mathbf{q}) = (\mathbf{r} - \mathbf{q})^2$. Let $\phi(s, q) = |\mathbf{r}_s - \mathbf{q}|$ and $f(x) = x^2, x > 0$.

$$\begin{aligned}
&g(\sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t)\mathbf{r}_s, \mathbf{q}) \\
=&[\sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t)\mathbf{r}_s - \mathbf{q}]^2 \\
=&[\sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t)\mathbf{r}_s \\
&- \sum_{t\in T}\sum_{s\in S} P(t|e)p(s|\mathcal{C}, t)\mathbf{q}]^2 \\
=&[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)|\mathbf{r}_s - \mathbf{q}|]^2 \\
=&f[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\phi(s, q)] \\
\propto&f(P(Y|do(C = \mathcal{C}), E = e, Q = q))
\end{aligned}$$

D.2 Relation Network

Let $g(\mathbf{r}, \mathbf{q}) = F[\mathbf{r} \oplus \mathbf{q} \oplus |\mathbf{r} - \mathbf{q}|]$. Let $\phi(s, q) = g(\mathbf{r}_s, \mathbf{q})$ and $f(x) = x$

$$\begin{aligned}
&g(\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s, \mathbf{q}) \\
=&F[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s \oplus \mathbf{q} \\
&\oplus |\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s - \mathbf{q}|] \\
=&F[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s \\
&\oplus \sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{q} \\
&\oplus |\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s \\
&- \sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{q}|] \\
\approx&F[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{r}_s \\
&\oplus \sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\mathbf{q} \\
&\oplus \sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)|\mathbf{r}_s - \mathbf{q}|] \\
=&\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)g(\mathbf{r}, \mathbf{q}) \\
=&f[\sum_{t\in T}\sum_{s\in S} P(t|e)P(s|\mathcal{C}, t)\phi(s, q)] \\
\propto&f(P(Y|do(C = \mathcal{C}), E = e, Q = q))
\end{aligned}$$

Here, we assume that the feature representations of the same event type in support are close to each other so that $|\sum_s p_s\mathbf{r}_s - \sum_s p_s q| \approx \sum_s p_s|\mathbf{r}_s - q|$.

E Implementation Details

All of our experiments are implemented on one Nvidia TITAN RTX. Our implementation is based on HuggingFace's Transformers (Wolf et al., 2019) and Allennlp (Gardner et al., 2018). We tune the hyperparameters based on the dev performance. We train each model 5 times with different random seed, and when evaluating, we sample 4 different support sets.

Metric-based Methods The hyperparameter is shown in Table 5. During training, the support set and the query is sampled in training set, the query contains 2 positive instances and 10 negative instances (5 times of positive instances). During validating, the support set and the query is sampled in dev set, the query contains 10 positive instances and 100 negative instances (10 times of positive

Model	ACE05		MAVEN		KBP17	
	Macro	Micro	Macro	Micro	Macro	Micro
Prototypical Network						
FS-Base(Snell et al., 2017)	66.2±3.8	63.8±4.1	44.1±1.6	44.0±2.3	67.1±1.7	68.0±1.6
FS-Lexfree (Lu et al., 2019)	50.4±2.8	50.7±3.6	24.9±1.1	20.5±1.4	60.8±2.7	60.4±3.7
FS-Cluster (Lai et al., 2020)	69.9±1.9	67.3±2.2	43.8±1.4	43.6±1.5	67.6±2.4	68.7±2.6
FS-Causal (Ours)	**76.8**±0.6	**76.3**±0.7	**51.8**±0.5	**55.1**±0.4	**72.6**±0.9	**74.9**±0.9
Relation Network						
FS-Base(Sung et al., 2018)	65.7±3.9	66.9±2.1	51.0±1.1	55.6±1.6	66.8±2.2	71.1±2.0
FS-Lexfree (Lu et al., 2019)	59.1 ±6.4	59.6±3.6	42.9±1.0	45.4±2.1	63.5±1.9	65.7±2.0
FS-Cluster (Lai et al., 2020)	54.4±2.9	57.2±3.2	45.8±2.3	51.4±1.4	57.5±4.2	62.0±3.2
FS-Causal (Ours)	**65.0**±2.1	**69.3**±1.5	**53.6**±0.7	**57.9**±1.0	**66.9**±0.1	**72.7**±0.9

Table 3: F1 score of 5-shot FSED on dev set. ± is the standard deviation of 5 random training rounds.

	ACE	MAVEN	KBP
Proto (support)	**76.25**	**55.11**	74.80
Proto (query)	65.75	37.89	69.25
Proto (support + query)	74.03	51.19	**74.93**
Relation (support)	68.40	54.34	69.09
Relation (query)	66.31	**57.85**	71.59
Relation (support + query)	**69.31**	56.93	**72.65**

Table 4: Micro F1 score of 5-shot FSED on dev set. (support) means the backdoor adjustment is used in the support set, (query) means the backdoor adjustment is used in the query.

	ACE	MAVEN	KBP
Optimizer	AdamW	AdamW	AdamW
Learning rate	2e-5	2e-5	2e-5
Warmup step	40	240	50
Batch size	1	1	1
patience	15	15	15
max epoch num	80	80	80
batches per epoch	40	240	50
λ for $P(T\|C)$	0.5	0.5	0.5
FS-Causal (Prototypical)	S	S	S+Q
FS-Causal (Relation)	S+Q	Q	S+Q

Table 5: Hyperparameters of metric-based methods. For FS-Causal, S means the backdoor adjustment is used in the support set, Q means the backdoor adjustment is used in the query.

	ACE	MAVEN	KBP
Optimizer	AdamW	AdamW	AdamW
Learning rate (Pretrain)	1e-5	1e-5	1e-5
batches per epoch (Pretrain)	50	200	200
Warmup step (Pretrain)	50	200	200
Batch size (Pretrain)	128	128	128
patience (Pretrain)	10	10	10
max epoch num (Pretrain)	50	50	50
Learning rate (Finetune)	2e-5	2e-5	2e-5
Learning rate (Finetune*)	1e-3	1e-3	1e-3
Finetuning Step (Finetune)	20	20	20
Finetuning Step (Pretrain + Finetune)	10	10	10

Table 6: Hyperparameters of finetuning-based methods.

instances). The results of dev set are shown in Table 3. For FS-Causal, we found that there is an impact on whether backdoor adjustment is applied separately to the support set and query, as shown in Table 4. Based on the best results of the dev set, we evaluate it on the test set.

Finetuning-based Methods The hyperparameter is shown in Table 6. For pretraining, we train a supervised event detection model using the training set. For finetuning, we use the support set to finetune the parameters of the event detection model and then detect the event in query.