

MOZILA: Continual Event Detection Through the Lens of Multi-Objective Optimization and Language Model Head Preservation*

Nam Le Hai, Linh Ngo Van**, and Sang Dinh

School of Information and Communications Technology, Hanoi University of Science and Technology, Vietnam

{namlh, linhnv, sangdv}@soict.hust.edu.vn

Continual Event Detection (CED) aims to identify and classify event triggers in evolving ontologies, where novel event types emerge incrementally. Unlike standard continual learning, CED faces highly imbalanced, long-tail distributions, semantic overlaps between event types, and a prevalence of negative spans, all of which exacerbate catastrophic forgetting. To address these challenges, we propose MOZILA, a novel framework that integrates Sharpness-Aware Multi-Objective Optimization (SaMOO) and Language Model Head Preservation (LMHP). SaMOO adapts gradient-based multi-objective optimization with sharpness-aware minimization to balance competing objectives, navigate the Pareto front, and locate flatter minima, enhancing robustness across sequential tasks. Meanwhile, LMHP strengthens representation learning by preserving essential linguistic knowledge from the pretrained language model head through distribution alignment, explored with three alternative strategies. To further address data imbalance, we leverage a generative model to synthesize event trigger representations, mitigating replay buffer limitations. Comprehensive experiments on standard CED benchmarks demonstrate that MOZILA consistently improves performance, achieving significant improvements over baseline methods. Ablation studies and error analyses highlight the individual contributions of each component and reveal key challenges in CED, including class imbalance and distinguishing similar event types across tasks. Our framework provides a robust and generalizable solution for continual event detection in dynamic, real-world environments. All resources are publicly available at:  MOZILA.

* A part of this work appears in Dao et al. (2024) and Le et al. (2024).

** Corresponding author.

Action Editor: Eduardo Blanco. Submission received: 19 April 2025; revised version received: 23 January 2026; accepted for publication: 10 March 2026.

<https://doi.org/10.1162/COLLa.617>

1. Introduction

Event Detection (ED) is a core task in Natural Language Processing (NLP) (Li et al. 2022; Chen et al. 2024; Nguyen, Ngo, and Nguyen 2023; Man et al. 2022), aiming to identify and classify event triggers into predefined types under a fixed ontology. Continual Event Detection (CED) extends this setting to a more realistic and challenging scenario, where the event ontology evolves over time as new event types are incrementally introduced. While dynamic label spaces have been widely studied in standard continual learning (CL), CED differs in several fundamental aspects. First, evaluation in CED is ontology-accumulative and asymmetric: At each stage, the model must recognize all previously learned event types, while future unseen types are absorbed into a dominant negative class (Not-Any [NA]). This leads to a **label reinterpretation** phenomenon, where trigger spans previously labeled as NA may later correspond to newly introduced event types, requiring the model to remain flexible rather than simply preserving past decisions. Second, CED naturally exhibits extreme long-tail distributions and severe class imbalance, with newly added event types often having very few instances and sentences containing a large number of negative spans. Finally, event types frequently exhibit semantic overlap, causing forgetting to manifest as increased confusion among related events rather than complete loss of earlier classes. Together, these properties make catastrophic forgetting in CED qualitatively different and more persistent than in standard class-incremental learning, motivating the need for continual learning strategies that can simultaneously preserve linguistic knowledge, handle imbalance, and maintain fine-grained semantic discrimination.

To address the challenge of catastrophic forgetting, several CL strategies have been proposed, broadly categorized into regularization-based (Ahn et al. 2019; Sun, Mu, and Hua 2023; Zhao et al. 2024), architecture-based (Li et al. 2019; Wang et al. 2023; Zhou et al. 2024), and memory-based approaches (Aljundi et al. 2018; Chaudhry et al. 2019; Bhat, Zonooz, and Arani 2023; Ma et al. 2024). Among existing approaches, the most effective strategies for CED combine memory-based replay with regularization techniques such as knowledge distillation. While regularization helps stabilize representations, the memory component plays a more pronounced role, as retaining a small fraction of past data in a *replay buffer* allows the model to directly revisit prior knowledge while learning new event types (Cao et al. 2020; Yu, Ji, and Natarajan 2021; Liu, Chang, and Huang 2022). Nonetheless, memory-based methods are not without limitations. First, when using these techniques in CL scenarios, managing the multiple objectives associated with both previous and current tasks becomes a critical challenge. Existing studies often aggregate these objectives using simple summation, which overlooks the complex and often conflicting trade-offs involved in optimizing for multiple tasks. Second, memory-based methods typically retain only a small subset of data from earlier tasks, which serve as representative examples to preserve learned knowledge while training of new event data. While effective to some extent, this limited number of samples can significantly reduce the model's representational capacity, potentially leading to suboptimal performance in recognizing both old and new events. To overcome these challenges, we propose MOZILA—a framework for CED that focuses on two key aspects: Multi-Objective Optimization and Representation Enhancement through Language Model Head Preservation.

Specifically, to effectively manage the multiple objectives arising from both old and new tasks, we explore the potential application of Multi-Objective Optimization (MOO) methods. In this context, gradient-based frameworks for MOO (Sener and Koltun 2018; Navon et al. 2022), which aim to identify solutions on the Pareto front, have emerged as

some of the most successful approaches. Although these methods have demonstrated significant success in static environments, their effective adaptation and application to the domain of CED remain largely unexplored. However, our investigation into the use of MOO techniques in the context of CED reveals notable challenges when these methods are directly applied without adaptation. In particular, while MOO frameworks are designed to balance competing objectives effectively with fixed tasks, their naive application to CED often results in significant performance degradation. This performance degradation can be attributed to two key factors. One key issue is the highly imbalanced distribution of training data in later tasks, where classes from the current task significantly outweigh those from earlier tasks in the replay buffer, leading to poor model generalization if not properly addressed (Johnson and Khoshgoftaar 2019). Additionally, existing MOO methods lack a clear criterion for identifying an optimal solution from the Pareto front that effectively mitigates catastrophic forgetting, as well as a systematic approach for achieving such a solution. In the CL context, where tasks are introduced incrementally rather than simultaneously, a robust solution must exceed traditional Pareto-optimality criteria by demonstrating the ability to adapt to new tasks while maintaining performance on previously learned ones. Several studies have highlighted the relationship between solutions located at flat minima in the loss landscape and their effectiveness in mitigating catastrophic forgetting (Mirzadeh et al. 2020; Hai et al. 2024). The concept of **flat minima** refers to regions of the loss function characterized by wide, flat basins, which are associated with more robust models and a lower risk of overfitting. Building on this insight, recent work (Phan et al. 2023) has explored the application of Sharpness-Aware Minimization (SAM) (Foret et al. 2021) for identifying Pareto-optimal solutions in flatter regions of the loss landscape, thereby enhancing multi-objective optimization. However, applying SAM in the context of CL requires careful consideration to fully realize its benefits. Specifically, it is essential to address the inherent instability of SAM and mitigate any adverse effects it may have on the model’s performance.

On the other hand, a significant challenge in memory-based approach for CED arises from the limited capacity of the replay buffer, which might discard valuable features from earlier tasks, especially for long tail event types. This results in incomplete information of past knowledge, weakening the model’s representation ability. Furthermore, the conventional practice of discarding the original language modeling head of pre-trained language models during fine-tuning exacerbates this issue. The language modeling head contains essential linguistic information, such as vocabulary distributions from contextual representations, which is crucial for effective alignment of the classifier module. However, state-of-the-art approaches (Yu, Ji, and Natarajan 2021; Liu, Chang, and Huang 2022; Qin et al. 2024; Wang, Wang, and Hu 2023) often train the classifier in isolation, ignoring this foundational knowledge. This misalignment reduces the efficiency of fine-tuning, sacrifices linguistic nuances, and increases the risk of overplasticity and catastrophic forgetting. Addressing these weaknesses in representation and alignment is crucial for improving the robustness of CED systems.

Contributions. From those perspectives, we introduce a novel framework MOZILA for CED. Our framework provides robust and significant improvements, achieving a 5–6% increase in F1 score over baseline methods. We conduct a comprehensive investigation of the proposed components; Language Model Head Preservation (LMHP), Sharpness-Aware Multi-Objective Optimization (SaMOO), and Synthetic Representation, demonstrating their individual and combined effectiveness. Additionally, we perform an error analysis that highlights key challenges in CED, including the imbalance of the NA class

and the difficulty of distinguishing similar event types across tasks. In summary, our article makes the following contributions:

1. We explore a wide range of MOO techniques in the context of CED, providing valuable insights. Our investigation reveals that naively applying these techniques can degrade performance in continual learning settings.
2. We propose a novel approach, SaMOO, that effectively integrates multi-task learning frameworks with SAM for CED. This adaptation is specifically designed to address the challenges of sequential task emergence in continual learning.
3. To improve representation ability, we introduce an innovative method that preserves essential linguistic information from the language modeling head. This is achieved through a family of distribution alignment techniques using Kullback-Leibler (KL) divergence, Mutual Information Maximization (MIM), and Optimal Transport (OT).
4. To tackle data imbalance and limited replay capacity, we propose leveraging a generative model to learn the underlying distribution of event trigger representations. This model synthesizes data to mitigate the imbalance between past-task and current-task data during replay.

Structure. The article is organized into five sections. Section 2 provides an overview of the background and related work relevant to CED. Section 3 introduces the components of the proposed framework, detailing its design and functionality. In Section 4, we present an empirical evaluation of our method, comparing its performance with existing CED approaches and conducting ablation studies to demonstrate the contribution of each component. Finally, Section 5 outlines potential directions for future work and concludes the article.

Key Extensions Over Prior Works. This work builds upon the prior studies of Dao et al. (2024) and Le et al. (2024), introducing several substantial extensions that advance their methodologies and broaden their applicability.

1. **Unified Framework:** We unify two previously distinct approaches into a general framework called MOZILA, which comprises two key components: LMHP and SaMOO. Our additional experiments demonstrate synergistic effects between LMHP and SaMOO, resulting in state-of-the-art performance on CED benchmarks.
2. **Broader MOO Scope:** While Le et al. (2024) explored only three MOO methods, this work significantly broadens this scope, evaluating five recent MOO techniques. We show that advanced methods like FairGrad can further enhance performance, offering a more comprehensive understanding of the design space.

3. **Generalized Alignment Strategies:** Whereas Dao et al. (2024) focused specifically on OT for alignment, we explore a broader family of distribution alignment strategies, including additional KL divergence and MIM. Our experiments reveal the effectiveness of MIM and the computational efficiency of KL, providing a nuanced comparative analysis.
4. **Substantial Experimental Expansion:** We conduct extensive new experiments that go beyond those in previous work. In addition to the standard CED benchmarks, we substantially expand the experimental scope by incorporating two additional and more challenging datasets, RAMS and GENEVA, which feature long, information-dense sentences, long-tailed event distributions, and fine-grained event ontologies. Furthermore, we perform detailed ablation studies, cross-method evaluations, forgetting, and computational discussion, providing deeper insights into each component’s contribution to the overall framework. Additionally, we present an error analysis of our framework, offering insights into its limitations that can be valuable for practitioners. We believe these contributions to represent a meaningful advancement over prior work, offering both theoretical insights and practical benefits for the field of CED in particular, and for real-time information extraction more broadly. Finally, our extensive empirical analysis provides new insights that can guide future research and encourage broader adoption in both academic and applied settings.

2. Background

2.1 Continual Learning

Continual Learning (CL), also known as lifelong learning (Chen and Liu 2018), is a paradigm in machine learning that aims to enable models to learn and adapt to new tasks or domains over time while retaining performance on previously learned tasks. Unlike traditional machine learning approaches, where models are trained on a fixed dataset and evaluated in a static environment, CL addresses dynamic scenarios where data arrives sequentially, reflecting real-world applications such as autonomous systems, robotics, and natural language processing. The primary challenge in CL is mitigating *catastrophic forgetting* (Thrun and Mitchell 1995; French 1999), a phenomenon in which the model’s performance on earlier tasks deteriorates as it learns new tasks. This challenge arises because the model’s parameters are updated during training on new data, often overwriting information critical to prior tasks. Overcoming catastrophic forgetting is central to ensuring the long-term viability and robustness of CL systems.

To address these challenges, researchers have proposed various strategies, which can be broadly categorized into three main approaches:

- Regularization-based methods (Kirkpatrick et al. 2017; Nguyen et al. 2017; Aljundi et al. 2018; Ahn et al. 2019; Sun, Mu, and Hua 2023; Zhao et al. 2024): These approaches introduce constraints during training to preserve knowledge of previously learned tasks. For instance, they penalize important parameters that contribute significantly to the earlier tasks, thereby maintaining stability.

- Architecture-based methods (Aljundi, Chakravarty, and Tuytelaars 2017; Serra et al. 2018; Li et al. 2019; Wang et al. 2023; Zhou et al. 2024): These techniques involve dynamically adapting the model’s architecture by adding new parameters or modules to accommodate new tasks. This ensures the separation of learned representations and minimizes interference between tasks.
- Memory-based methods (Lopez-Paz and Ranzato 2017; Rebuffi et al. 2017; Chaudhry et al. 2019; Bhat, Zonooz, and Arani 2023): These methods explicitly store a subset of past data or task representations in a memory buffer. By incorporating this data during the training of new tasks, these approaches gain access to representative samples from previous tasks, thereby reinforcing prior knowledge and mitigating forgetting.

Memory-based methods have proven to be the most effective among these approaches, particularly within the domain of lifelong natural language processing learning (de Masson D’Autume et al. 2019).

2.2 Event Detection

Traditional Event Detection. Event Detection (ED) (Wadden et al. 2019) is a critical task in NLP that involves identifying and classifying event triggers within a given text. For instance, in the sentence “*A woman died in the hospital*”, an ED system should identify a *Die* event and correctly associate it with the trigger word “*died*”. Traditional ED systems are designed to work with a predefined set of event classes (also referred to as event types) and rely on a one-time training process using a fixed dataset.

Formally, the training dataset $\mathcal{D}$ comprises text-label pairs $\mathcal{D} = \{(x_i, y_i)\}_1^M$, where each x_i consists of a L -token context $w_{1:L}$ and a trigger span (s, e) , which specifies the start and end positions of the event trigger. The y_i denotes the event type associated with the trigger, selected from a predefined set of event $\mathcal{Y}$, or NA if $w_{s:e}$ does not correspond to any known event type. ED systems commonly employ an encoder $\mathcal{M}$, such as pre-trained language models like BERT (Devlin et al. 2019), to generate contextualized representations of event triggers, which are then used for classification. Specifically, the model encodes the context $w_{1:L}$, and the representations of the start token w'_s and end token w'_e of the trigger span are concatenated to form a feature vector h . This feature vector is then passed through a multi-layer perceptron (MLP) for event type classification. The entire procedure can be represented by the following equations:

$$\begin{aligned} w'_{1:L} &= \mathcal{M}(w_{1:L}) \\ h &= \text{concat}(w'_s, w'_e) \\ p^{CED} &= \text{Softmax}(\text{MLP}(h)) \end{aligned} \quad (1)$$

Let $\mathcal{D}^{train} = \{x_i, y_i\}_1^m$ denote the training dataset, the training process utilizes the cross-entropy loss function.

$$\mathcal{L}_{ce}(\mathcal{D}^{train}) = -\frac{1}{m} \sum_{i=1}^m \sum_{c=1}^C y_{i,c} \log(p_{i,c}) \quad (2)$$

where m is the number of training samples, C is the number of event classes, $y_{i,c} \in \{0, 1\}$ is the one-hot ground truth label of sample i for class c , and $p_{i,c}$ is the predicted probability of sample i belonging to class c .

However, it is important to note that there is an imbalance between the data of the “unknown” type (i.e., NA) and other event types. To mitigate the issue of NA-labeled instances dominating the overall gradient, the loss function is regularized by incorporating a hyperparameter coefficient η :

$$\mathcal{L}_{ed} = \frac{\eta}{|\mathcal{D}_{NA}^{train}|} \mathcal{L}_{ce}(\mathcal{D}_{NA}^{train}) + \frac{1 - \eta}{|\mathcal{D}_{non-NA}^{train}|} \mathcal{L}_{ce}(\mathcal{D}_{non-NA}^{train}) \quad (3)$$

in which $\mathcal{D}_{NA}^{train}$ denotes the set of all NA-label instances in the training set $\mathcal{D}^{train}$ and $\mathcal{D}_{non-NA}^{train} = \mathcal{D}^{train} / \mathcal{D}_{NA}^{train}$.

Continual Event Detection. In real-world scenarios, new event types emerge continuously. To address this challenge, the concept of CED has been introduced (Cao et al. 2020; Yu, Ji, and Natarajan 2021). CED can be viewed as a subfield of continual information extraction, which studies how information extraction systems adapt to newly emerging schemas and knowledge over time without forgetting previously learned information (Wang, Wang, and Hu 2023; Dao et al. 2024; Qin et al. 2024; Tran et al. 2024; Pham et al. 2025; Le et al. 2025; Dao et al. 2026). In particular, CED involves a sequence of ED tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$ that arise progressively, requiring models to adapt to newly introduced events while preserving their ability to recognize previously learned ones, as illustrated in Figure 1.

Consider a sequence of datasets $\{\mathcal{D}_t\}$, where each $\mathcal{D}_t$ comprises training, validation, and testing subsets ($\mathcal{D}_t^{train}, \mathcal{D}_t^{val}, \mathcal{D}_t^{test}$) along with a specific set of event classes $\mathcal{Y}_t$ corresponding to the current time step t . Any two event class sets are disjointed, such that $\mathcal{Y}_t \cap \mathcal{Y}_{t'} = \emptyset \quad \forall t \neq t'$. At stage t , the model is required to detect events based on the combined ontology of all previously encountered event types, defined as $\mathcal{O}_t = \bigcup_{i=1}^t \mathcal{Y}_i$.

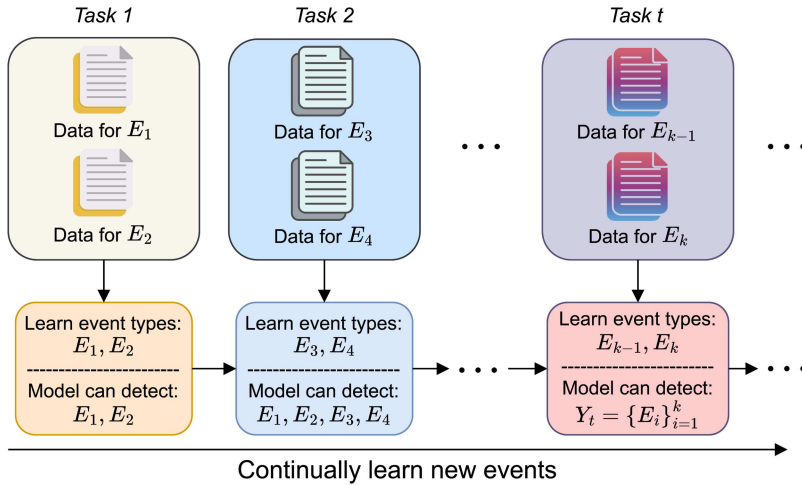


Figure 1

Continual Event Detection: A model continuously learns new event classes as corresponding data becomes available. At each task t , the model retains the capability to detect all event types encountered up to this task.

The NA label represents an undefined category and is included in every task. At step t , the event detection model updates its parameters using the training data $\mathcal{D}_t^{\text{train}}$. The updated model is expected to maintain strong performance across all previously encountered event classes. During the testing phase at step t , the model's performance is evaluated on the testing data of all observed classes up to that point, represented as $\bigcup_{i=1}^t \mathcal{D}_i^{\text{test}}$.

Several studies (Cao et al. 2020; Yu, Ji, and Natarajan 2021; Liu, Chang, and Huang 2022) have demonstrated the effectiveness of replay-based methods in addressing catastrophic forgetting in CED. These methods frequently use the herding algorithm (Welling 2009) to select representative data for each event class, excluding NA, for inclusion in the replay buffer. Memory-based training for CED models typically involves two key techniques: *knowledge distillation* and *knowledge transfer*. The following provides detail of these techniques, with R representing the replay buffer.

Knowledge distillation is a widely recognized technique for transferring knowledge between models. Let θ_{t-1} and θ_t represent the models before and after training on the t^{th} task, respectively. By passing an instance x stored in the memory buffer R through these two models, following the process outlined in Equation (1), we derive the probability distributions over the learned event types, p_{t-1} and p_t . Based on these distributions, the distillation loss is defined as

$$\mathcal{L}_d = - \sum_{x \in R} \sum_{c=1}^C p_{t-1}(x, c) \log(p_t(x, c)) \quad (4)$$

where R denotes the replay buffer, C is the number of event classes, $p_{t-1}(x, c)$ is the predicted probability of sample x belonging to class c under the previous model (from step $t-1$), and $p_t(x, c)$ is the corresponding probability under the current model.

Knowledge Transfer aims to align the prediction probability p_t of the current model with the prediction probability of the previous model, scaled by a temperature τ , represented as $q_{t-1} \sim (p_{t-1})^{1/\tau}$. To accomplish this, the knowledge transfer loss is formulated as

$$\mathcal{L}_{kt} = - \frac{1}{|\mathcal{D}'_t|} \sum_{x \in \mathcal{D}'_t} \sum_{c=1}^C q_{t-1}(x, c) \log(p_t(x, c)) \quad (5)$$

Here, $\mathcal{D}'_t$ represents the modified training set for the t^{th} task. As noted by Yu, Ji, and Natarajan (2021), instances assigned a high probability of the NA label by the previous model should be excluded from knowledge transfer, as these instances are less relevant to the previously learned event types. Consequently, rather than directly utilizing $\mathcal{D}_t$, we construct $\mathcal{D}'_t$ by filtering out instances where the probability output to the NA class by the previous model θ_{t-1} surpasses a predefined threshold, ensuring better alignment with the objectives of the knowledge transfer loss.

2.3 Multi-Objective Optimization

MOO (Caramia et al. 2020) provides a framework for optimizing multiple tasks simultaneously. In multi-task learning, each task's loss forms one component of an objective vector, and the goal is to find parameters that balance these objectives (Sener and Koltun

2018). Originating from Pareto’s work (Gunantara 2018), MOO focuses on achieving trade-offs rather than optimizing a single task.

Let θ represent the set of all model parameters within the feasible space Θ . For each task $\mathcal{T}_t$, we define a corresponding training loss function $\mathcal{L}_t(\theta)$, which measures the error or cost corresponding to the model’s predictions for a specific task $\mathcal{T}_t$. In the context of K tasks, the goal of MOO is to minimize all K loss functions concurrently. This objective can be expressed using the following formulation:

$$\min_{\theta} [\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \dots, \mathcal{L}_K(\theta)] \quad (6)$$

Given θ_1^* and θ_2^* as two feasible solutions to Equation (6), we state that θ_1^* dominates θ_2^* if and only if $\mathcal{L}_i(\theta_1^*) \leq \mathcal{L}_i(\theta_2^*) \forall i \in \{1, \dots, K\}$ and $\exists j \in \{1, \dots, K\}$ s.t. $\mathcal{L}_j(\theta_1^*) < \mathcal{L}_j(\theta_2^*)$. A feasible solution θ^* is considered **Pareto optimal** if there does not exist any other feasible solution that dominates θ^* . The set of Pareto-optimal solutions is called the **Pareto front**. In this context, the primary challenge arises from the potential misalignment and conflict among these loss functions, which can make simultaneous optimization difficult. A key characteristic of MOO is the absence of a natural linear ordering for vectors, making it generally impossible to directly compare solutions and identify an explicit optimal value. Thus, instead of a single global minimum, the solution to this optimization problem typically involves finding a Pareto optimal. Balancing these competing objectives is crucial for effective MOO, as it requires careful design and tuning of model architectures and training algorithms to achieve an optimal trade-off among all tasks.

Prominent studies (Yu et al. 2020; Liu et al. 2021a,b; Navon et al. 2022; Liu et al. 2024; Ban and Ji 2024; He et al. 2024) have utilized approaches based on the multiple-gradient Descent Algorithm for multi-objective optimization, with the objective of identifying solutions on the Pareto front. These methods share a common underlying principle: They compute the update direction for the model parameters using an aggregation algorithm $\mathcal{A}$ to combine gradients $g_i^{(\tau)} = \nabla_{\theta^{(\tau-1)}} \mathcal{L}_i(\theta)$ from individual tasks at training step s , expressed as $\Delta\theta = \mathcal{A}(g_1^{(s)}, \dots, g_K^{(s)})$. Specifically, the aggregation algorithms used in each method are discussed as follows.

- PCGrad (Yu et al. 2020): Modifies conflict gradients by projecting each onto the normal plane of the other, effectively removing the interfering components. Specifically, if g_i conflicts with g_j (cosine similarity is negative):

$$g_i^{(s)} = g_i^{(s)} - \frac{g_i^{(s)} \cdot g_j^{(s)}}{\|g_j^{(s)}\|^2} g_j^{(s)} \quad \text{then} \quad \Delta\theta = \sum_i g_i^{(s)} \quad (7)$$

- CAGrad (Liu et al. 2021a): Improves MTL by finding an update direction that maximizes the worst local improvement in a neighborhood of the average gradient.

- IMTL (Liu et al. 2021b): Learns the scaling factors in a way that the combined gradient of task-shared parameters has equal projections onto the raw gradients of each individual task.

$$\begin{aligned}
 w_{2:K} &= g_1^{(s)} U^\top (DU^\top)^{-1} \\
 w &= [1 - 1w_{2:K}^\top, w_{2:K}] \\
 \text{then } \Delta\theta &= \sum_i w_i^{(s)} g_i^{(s)}
 \end{aligned} \tag{8}$$

where $D^\top = [g_1^{(s)\top} - g_2^{(s)\top}, \dots, g_1^{(s)\top} - g_K^{(s)\top}]$ and $U^\top = [u_1^\top - u_2^\top, \dots, u_1^\top - u_K^\top]$ with $u_i = g_i^{(s)} / \|g_i^{(s)}\|$ are gradient differences and unit-norm gradient differences vectors.

- NashMTL (Navon et al. 2022): Combines the gradients at each step using the Nash bargaining solution (Nash 1953). Let $G^{(s)}$ be the matrix with columns $g_i^{(s)}$, and let w be the weight matrix used to combine the gradients. The weight matrix w is obtained by solving the following equation:

$$G^{(s)\top} G^{(s)} w = 1/w \quad \text{then } \Delta\theta = G^{(s)} w \tag{9}$$

- FairGrad (Ban and Ji 2024): Solves the α -fair MTL utility maximization problem, drawing inspiration from fair resource allocation in communication networks. Similar to NashMTL, the weight matrix w is derived by solving the following equation:

$$G^{(s)\top} G^{(s)} w = w^{-1/\alpha} \quad \text{then } \Delta\theta = G^{(s)} w \tag{10}$$

where $\alpha \in [0, 1) \cup (1, +\infty)$ denotes the parameter governing the α -fair framework.

- ExcessMTL (He et al. 2024): Adjusts task weights based on their distance to convergence, quantified by excess risks, which represent the loss gap between the current model and the optimal model within the hypothesis class. Specifically, excess risks is computed as:

$$\begin{aligned}
 \mathcal{E} &= g_i^{(s)\top} H_i^{(s)-1} g_i^{(s)}; \quad w_i^{(s)} = w_i^{(t-1)} \exp(\eta_w \mathcal{E}) \\
 \text{then } \Delta\theta &= \sum_i w_i^{(s)} g_i^{(s)}
 \end{aligned} \tag{11}$$

where $H_i^{(s)} \approx \text{diag}(\sum_{s=1}^t g_i^{(s)} g_i^{(s)\top})^{1/2}$ is the Hessian matrix, approximated using the empirical Fisher information matrix. η_w and s refer to the step size and training step.

While these methods are theoretically well-founded, their empirical effectiveness in the context of continual learning remains limited and insufficiently explored. In

continual learning, a viable solution not only satisfies the Pareto-optimal criterion, but also exhibits robustness in adapting to new, unseen tasks while maintaining stable performance on previously learned tasks, thereby mitigating catastrophic forgetting.

2.4 Optimal Transport

Optimal Transport (OT) is a mathematical framework designed to quantify the distance between two probability distributions, even when they are defined over different supports. Let $\mathbf{p}_c$ represent a distribution over a set of C classes, and $\mathbf{p}_v$ denote a distribution over a set of V classes. The OT distance between $\mathbf{p}_v$ and $\mathbf{p}_c$ is defined as the minimal cost required to transform $\mathbf{p}_v$ into $\mathbf{p}_c$, which can be expressed as:

$$d_M(\mathbf{p}_v, \mathbf{p}_c) := \min_{\mathbf{P} \in U(\mathbf{p}_v, \mathbf{p}_c)} \langle \mathbf{P}, \mathbf{M} \rangle \quad (12)$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product. The cost matrix $\mathbf{M} \in \mathbb{R}_{\geq 0}^{V \times C}$ represents the cost of transporting mass between classes in V and classes in C , with each entry m_{vc} indicating the cost of transporting from class $v \in V$ to class $c \in C$. The matrix $\mathbf{P} \in \mathbb{R}_{> 0}^{V \times C}$ is the transport plan, detailing how mass is moved between distributions, and $U(\mathbf{p}_v, \mathbf{p}_c)$ is the set of all feasible transport plans that match the marginal distributions $\mathbf{p}_v$ and $\mathbf{p}_c$. For instance, considering two discrete random variables $X \sim \text{Categorical}(\mathbf{p}_v)$ and $Y \sim \text{Categorical}(\mathbf{p}_c)$, the transport plan $\mathbf{P}$ corresponds to a joint probability distribution over (X, Y) , where $p(X = i, Y = j) = p_{ij}$. The set $U(\mathbf{p}_v, \mathbf{p}_c)$ thus encompasses all joint probability distributions that satisfy the marginal constraints, forming what is known as a transport polytope.

Directly optimizing the original OT problem in Equation (12) is computationally intensive and time-consuming. To mitigate this issue, Cuturi (2013) proposes an entropy-regularized OT distance, commonly referred to as the Sinkhorn distance:

$$s_M(\mathbf{p}_v, \mathbf{p}_c) := \min_{\mathbf{P} \in U(\mathbf{p}_v, \mathbf{p}_c)} \langle \mathbf{P}, \mathbf{M} \rangle - H(\mathbf{P}) \quad (13)$$

which introduces an entropy function serving as the regularizing term:

$$H(\mathbf{P}) = - \sum_{i,j} \mathbf{P}_{i,j} (\log(\mathbf{P}_{i,j} + 1)) \quad (14)$$

In most cases, the cost matrix $\mathbf{M}$ is not available. A common approach is to estimate $\mathbf{M}$ dynamically throughout the training process. The Sinkhorn distance introduces an entropic penalty to the transport plan. This formulation allows the problem to be solved efficiently with iterative matrix scaling, yielding nearly linear convergence and GPU-friendly parallelization (Cuturi 2013). As a result, the Sinkhorn distance offers a tractable and scalable approximation of OT, avoiding the prohibitive cost of exact solvers while maintaining sufficient accuracy for our alignment objective.

3. Methodology

3.1 Overview

In this section, we provide details on our proposed MOZILA, a memory-based framework for CED that prioritizes two key aspects:

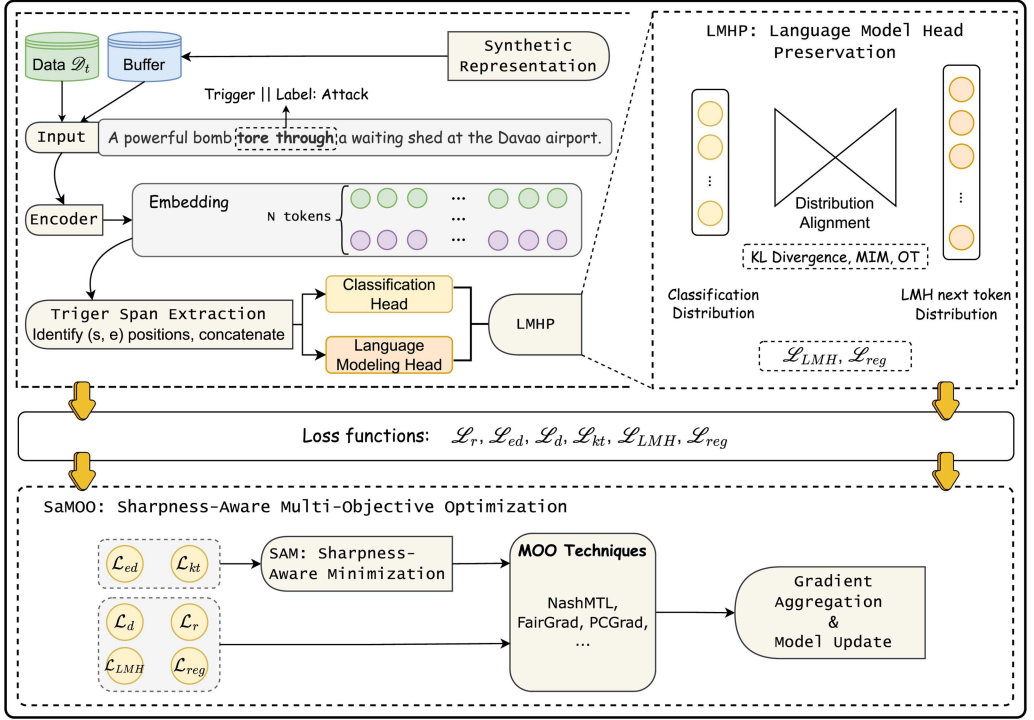
- **Representation Ability Enhancement (3.2):** Enhancing the model's capability to learn and retain informative knowledge, ensuring robustness in continual scenarios. Our approach focuses on preserving the rich linguistic information from the language model head, which is often discarded in previous work. By retaining this information, we not only help maintain knowledge within the backbone but also facilitate better alignment of the main classifier, ultimately improving representation learning.
- **Multiple Objectives Optimization (3.3):** Effectively balancing and optimizing multiple competing objectives between previous tasks stored in the replay buffer and the current task. Specifically, we leverage the benefits of flat minima for continual learning and introduce the use of SAM alongside MOO techniques to enhance the learning process.

Furthermore, to effectively accomplish these two objectives, we propose Data Balancing via Synthetic Representations, as detailed in Section 3.4. This approach not only mitigates the overfitting issue caused by the limited data retained from previous tasks in the memory buffer but also alleviates the imbalance in the training data distribution, thereby enhancing MOO and supporting the representation learning process. The overview of MOZILA framework is demonstrated in Figure 2 and Algorithm 1. In particular, our framework comprises two core modules: LMHP and SaMOO. The LMHP module is designed to retain the rich linguistic knowledge embedded in the pre-trained language model by aligning the classifier's output distribution with that of the language modeling head, thereby enhancing representation quality and mitigating information loss during fine-tuning. On the other hand, SaMOO integrates gradient-based MOO with sharpness-aware minimization to effectively balance competing objectives across tasks while guiding the model towards flatter minima in the loss landscape.

3.2 Learning with Language Model Head Preservation

The language modeling head encodes vital information about vocabulary distribution from contextual representations. Removing this head compromises essential linguistic nuances, hindering classifier alignment and efficient fine-tuning. In this section, we introduce the proposed distribution alignment method, which ensures that the probability distribution produced by the classifier head aligns with the inherent distributional properties of the Language Model Head (LMH). By mapping the softmax class probabilities from the classifier head to closely resemble the pre-trained distribution, this approach enables a smooth integration of task-specific knowledge while minimizing divergence from the model's original linguistic representations.

Specifically, after feeding input x into the encoder $\mathcal{M}$ to derive w'_s and w'_e , as described in Equation (1), this event trigger is then processed through a pre-trained

**Figure 2**

Overall architecture of MOZILA. The framework consists of two main modules: Language Model Head Preservation (LMHP), which aligns classification and LM head distributions, and Sharpness-Aware Multi-Objective Optimization (SaMOO), which integrates multiple loss objectives with SAM and MOO techniques for robust optimization. A Synthetic Representation module augments replay data to mitigate imbalance.

language model along with its original language modeling head, yielding a probability distribution over a vocabulary of $|V|$ words.

$$\begin{aligned}
 p_s^{LMH} &= \text{Softmax}(\text{LMH}(w'_s)/\tau) \\
 p_e^{LMH} &= \text{Softmax}(\text{LMH}(w'_e)/\tau) \\
 p^{LMH} &= (p_s^{LMH} + p_e^{LMH})/2
 \end{aligned} \tag{15}$$

where LMH is a pre-trained language model head of $\mathcal{M}$, τ is temperature coefficient, and $p^{LMH} \in R^{|V|}$ is distribution of the event trigger over dictionary. On the other hand, each event trigger corresponds to a probability distribution over C classes, denoted as $p^{CED} \in R^C$, where each entry represents the likelihood of the trigger being classified into a specific category within the ontology, as demonstrated in Equation (1). Given that p^{LMH} and p^{CED} are distributions with different supports for the same event trigger, we aim to train the model to push p^{CED} towards p^{LMH} . Specifically, we utilize KL divergence and OT to achieve alignment between these distributions.

Algorithm 1. Training Procedure for MOZILA in Task t **Input:**

Model parameters θ
 Replay buffer R , set of previous event types R_C
 Generative models $\mathcal{GM} = \{\text{GM}_c\}$ for each class c in R_C
 Training data D_t , Modified training data D'_t , Set of event types in task t C_t

Output:

Updated parameter θ^*

Optimization Procedure:

```

1:  $\tilde{R} \leftarrow \{\}$ 
2: for  $c \in R_C$  do
3:    $\tilde{R} \leftarrow \tilde{R} \cup \text{GM}_c.\text{sample}()$  ▷ Synthetic Representation
4:  $R_a = R \cup \tilde{R}$ 
5: Compute  $\mathcal{L}_r, \mathcal{L}_d$  on  $R_a$ , using Equations (32) and (33)
6: Compute  $\mathcal{L}_{ed}$  on  $D_t$  and  $\mathcal{L}_{kt}$  on  $D'_t$ , using Equations (3) and (5)
7: Compute  $\mathcal{L}_{LMH}$  using Equations (16, 23, 19) on  $R_a \cup D_t$  ▷ LMHP
8: Compute  $\mathcal{L}_{reg}$  using Equation (24) on  $R_a \cup D_t$ 
9:  $\theta^* \leftarrow \text{SaMOO.update}(\theta, \mathcal{L}_r, \mathcal{L}_d, \mathcal{L}_{ed}, \mathcal{L}_{kt}, \mathcal{L}_{LMH}, \mathcal{L}_{reg})$  ▷ SaMOO Algorithm 2
10: for  $c_t \in C_t$  do
11:   Train  $\text{GM}_{c_t}$  using  $D_t$ 
12:    $\mathcal{GM} \leftarrow \mathcal{GM} \cup \{\text{GM}_{c_t}\}$  ▷ Update new generative model

```

Language Model Head Preservation via KL Divergence. To align the probability distributions p^{LMH} from the LMH and p^{CED} from the ED classifier, we first apply a linear projection function f_ϕ to map p^{CED} into the same dimensional space as p^{LMH} . Then, we minimize the KL divergence between the two distributions:

$$\mathcal{L}_{LMH} = \mathcal{L}_{KL} = KL(p^{LMH} || f_\phi(p^{CED})) = p^{LMH} \log \left(\frac{p^{LMH}}{f_\phi(p^{CED})} \right) \quad (16)$$

By incorporating this loss, we encourage the classifier's output to align with the distributional properties of the pre-trained language model, preserving essential linguistic information while fine-tuning for event detection.

Language Model Head Preservation via Mutual Information. Inspired by Guo, Liu, and Zhao (2022), our objective is to maximize Mutual Information (MI) between the LMH representations and those from the event classifier, formulated as $MI = I[p^{CED}, p^{LMH}]$. According to van den Oord, Li, and Vinyals (2018):

$$MI \geq \log B + \text{InfoNCE}(\{x_i\}_{i=1}^B; h) \quad (17)$$

where we have defined

$$\begin{aligned} \text{InfoNCE}(\{x_i\}_{i=1}^B; h) &= \frac{1}{B} \sum_{i=1}^B \log \frac{h(p_i^{\text{CED}}, p_i^{\text{LMH}})}{\sum_{j=1}^B h(p_i^{\text{CED}}, p_j^{\text{LMH}})} \\ h(p_i^{\text{CED}}, p_j^{\text{LMH}}) &= \exp \frac{p_i^{\text{CED}T} W p_j^{\text{LMH}}}{\tau} \end{aligned} \quad (18)$$

where τ is the temperature, B is mini-batch size, and W is a trainable parameter. Then, the MI loss function in our implementation is:

$$\mathcal{L}_{\text{LMH}} = \mathcal{L}_{\text{MI}} = - \sum_{(x_i, y_i) \in D_i^{\text{train}}} \text{InfoNCE}(\{x_i\}_{i=1}^B; h) \quad (19)$$

Language Model Head Preservation via Optimal Transport. As discussed in Section 2.4, OT provides a framework for measuring the distance between two probability distributions, even when they are defined over different supports or exist in distinct dimensional spaces. Following Equation (12), we define the OT-based alignment for p^{LHM} and p^{CED} as:

$$d_{\mathbf{M}}(p^{\text{LHM}}, p^{\text{CED}}) := \min_{\mathbf{P} \in U(p^{\text{LHM}}, p^{\text{CED}})} \langle \mathbf{P}, \mathbf{M} \rangle \quad (20)$$

where $\mathbf{M} \in \mathbb{R}_{\geq 0}^{V \times C}$ represents the cost matrix, $\mathbf{P} \in \mathbb{R}_{> 0}^{|V| \times C}$ denotes the transport plan, and $U(p^{\text{LHM}}, p^{\text{CED}})$ is defined as the set of all feasible transport plans. Directly optimizing this equation poses a time-consuming challenge. Hence, we adopt Sinkhorn distance as demonstrated in Equation (21):

$$s_{\mathbf{M}}(p^{\text{LHM}}, p^{\text{CED}}) := \min_{\mathbf{P} \in U(p^{\text{LHM}}, p^{\text{CED}})} \langle \mathbf{P}, \mathbf{M} \rangle - \mathbf{H}(\mathbf{P}) \quad (21)$$

where $\mathbf{H}(\mathbf{P}) \stackrel{\text{def}}{=} - \sum_{i,j} \mathbf{P}_{ij} (\log(\mathbf{P}_{ij} - 1))$ is the entropy function of the transport plan. Since $\mathbf{M}$ is unavailable, we enhance the learning process by constructing $\mathbf{M}$ using cosine similarity, formulated as follows:

$$m_{vc} = 1 - \cos(\mathbf{e}_v, \mathbf{e}'_c) \quad (22)$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity function, and $\mathbf{e}_v \in \mathbb{R}^D$ and $\mathbf{e}'_c \in \mathbb{R}^D$ are the embedding vectors with D dimensions corresponding to class v and class c , respectively, derived from the learnable matrices $E_V \in \mathbb{R}^{|V| \times D}$ and $E_C \in \mathbb{R}^{C \times D}$. These matrices are optimized by minimizing the Sinkhorn distance described in Equation (21).

Several studies (Frogner et al. 2015; Zhao et al. 2021) further suggested combining the OT loss with a conventional cross-entropy loss to better guide the model. By parameterizing $\mathbf{M}$ with E_c , the collection of class embeddings, the final OT objective function is expressed as:

$$\mathcal{L}_{\text{LMH}} = \mathcal{L}_{\text{OT}} = \min_{\theta, E_c} [s_{\mathbf{M}}(p^{\text{LHM}}, p^{\text{CED}}) - \epsilon p^{\text{LHM}} \log \phi(p^{\text{CED}})] \quad (23)$$

where $\phi(z) := \text{Softmax}((2 - \mathbf{M})z)$ and ϵ is a hyperparameter that regularize the weight of the expected likelihood. Additionally, given the large vocabulary size, which includes numerous subwords and unrelated terms, we reduce computational complexity by selecting a subset of words consisting only of verbs. In each batch, this subset is combined with the tokens present in the batch to compute the OT loss.

Class Embedding Regularization. To maintain the consistency of class representations across tasks, an additional regularization term enforces the proximity of class representations in the current task to those in the most recent task:

$$\mathcal{L}_{reg} = \|E_C^{(t)} - E_C^{(t-1)}\|^2 \quad (24)$$

Final Objective. Finally, the overall objective function is formulated by integrating the loss functions from Equations (2), (3), (4), (5), (16), (23), and (24), as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{ed} + \mathcal{L}_d + \mathcal{L}_{kt} + \mathcal{L}_{LMH} + \gamma \mathcal{L}_{reg} \quad (25)$$

where γ is the regularization coefficient.

Comparison with Regularization-Based Approach. Although LMHP can be viewed as a form of regularization, it is conceptually distinct from conventional regularization-based continual learning techniques such as knowledge distillation or EWC-style constraints. Existing methods primarily regularize *task-specific behavior* by constraining parameter updates or output distributions to preserve performance on previously seen tasks. In contrast, LMHP introduces an *additional and orthogonal regularization signal* that preserves the *general linguistic knowledge* encoded in the pretrained language model head, which is task-agnostic and independent of task order. Importantly, LMHP is not designed to replace existing regularization mechanisms. As shown in Equation (25), our framework already incorporates standard regularization-based components (i.e., exemplar replay, distillation, and knowledge transfer) to retain task-specific knowledge across tasks. LMHP complements these components by anchoring the event classifier head to the frozen pretrained LM head via projection and OT, thereby injecting vocabulary-level semantic priors that are absent in conventional regularization approaches.

3.3 Multi-Objective Optimization with Sharpness-Aware Sequential Learning

As presented in Sections 2.2 and 3.2 and Equation (25), the training process of our model can be viewed as a MOO problem, with the goal of simultaneously minimizing six objectives: $\mathcal{L}_{ce}$, $\mathcal{L}_{ed}$, $\mathcal{L}_d$, $\mathcal{L}_{kt}$, $\mathcal{L}_{LMH}$, and $\mathcal{L}_{reg}$. These loss functions correspond to different data utilizations, such as replay data from previous tasks or data from the current task, as outlined in Equations (2), (3), (4), and (5), respectively. Note that $\mathcal{L}_{ce}$ and $\mathcal{L}_d$ are used as the replay loss $\mathcal{L}_r$ (32) and the modified distillation loss (33), respectively, with the modified replay training buffer in the final loss function. Afterward, we can effectively tackle this MOO problem by utilizing multi-task learning (MTL) frameworks, as outlined in Section 2.3.

Despite their promise, existing MTL frameworks have shown limited empirical performance (Phan et al. 2023). This limitation stems partly due to their focus on minimizing empirical error while overlooking the geometric properties of the loss landscape. To address this, Phan et al. (2023) introduced an approach that enhances the

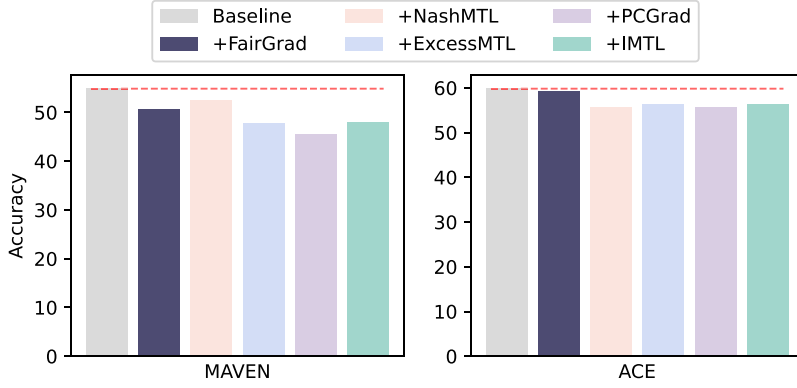


Figure 3

The accuracy after training the final task (Task 5) directly using MOO techniques on the Baseline model for MAVEN and ACE datasets. Here, the Baseline refers to the KT method, which is detailed in Section 4. The degradation is clearly observed when MOO methods are applied naively, indicating their limitations in the CED setting.

robustness of multi-task models by exploring task-based flat regions using SAM. The benefits of flat minima have also been demonstrated in the continual learning setting, where they help mitigate catastrophic forgetting (Mirzadeh et al. 2020; Hai et al. 2024). However, CED and continual learning problems in general—differ fundamentally from traditional MTL, as tasks do not appear simultaneously but sequentially. As a result, directly applying the MOO methods to this setting is suboptimal and may even degrade performance (see Figure 3).

Sharpness-Aware Multi-Objective Optimization (SaMOO). The conventional training approach focuses solely on minimizing empirical loss, which often results in overfitting, limiting the model’s ability to generalize to unseen data. To address this issue, Foret et al. (2021) propose an alternative strategy that minimizes the worst-case loss rather than directly optimizing the training loss. This reformulation leads to the following optimization problem:

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \quad (26)$$

where $\|\cdot\|_2$ is the l_2 norm and ρ is the radius of the neighborhood; and $\boldsymbol{\theta}$ denotes the model’s parameters. To solve the inner maximization in problem (26), Foret et al. (2021) first estimate $\boldsymbol{\epsilon}$ by using the first-order Taylor expansion to estimate $\mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})$.

$$\begin{aligned} \boldsymbol{\epsilon}^* \in \operatorname{argmax}_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) &\approx \operatorname{argmax}_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \boldsymbol{\epsilon}^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \\ &\approx \rho \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})\|_2} \end{aligned} \quad (27)$$

Once ϵ is approximated, the gradient of worst-case loss will be used to update the parameter θ :

$$\mathbf{g}^{SAM} := \nabla_{\theta} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}(\theta + \epsilon) \approx \nabla_{\theta} \mathcal{L}(\theta + \epsilon)|_{\theta + \epsilon^*} \quad (28)$$

After obtaining the task-specific gradients $\{\mathbf{g}_i^{SAM}\}_{i=1}^K$, we can leverage one of the gradient-based MTL frameworks introduced in Section 2.3 to address our modified MOO problem. Specifically, the aggregation method $\mathcal{A}$ is applied to these gradients to compute the update value $\mathcal{A}(\mathbf{g}_1^{SAM}, \dots, \mathbf{g}_K^{SAM})$. Intuitively, SAM seeks solutions that remain robust under small parameter perturbations by explicitly penalizing sharp minima. Equation (27) shows that the worst-case perturbation ϵ^* can be efficiently approximated in the direction of the gradient, scaled by the perturbation radius ρ . This avoids solving the maximization problem exactly, which would be computationally prohibitive. Then, as shown in Equation (28), the parameter update is guided by the gradient of the perturbed loss, effectively encouraging the optimizer to prefer flatter regions of the loss landscape. This two-step approximation strikes a balance between robustness and efficiency, making SAM suitable for continual learning where stability across tasks is critical.

SaMOO Sequential Learning for CED. Given that after training on task t , the obtained solution satisfies the flat-minima criterion, it follows that when progressing to task $t + 1$, the solution already resides within the flat regions of the loss landscapes for the replay loss ($\mathcal{L}_r$) and distillation loss ($\mathcal{L}_d$). Applying SAM to those past-task losses might introduce unnecessary perturbation and noise, which may disrupt previously acquired knowledge. Therefore, to minimize interference and preserve stability in continual learning, we restrict SAM to the event detection loss ($\mathcal{L}_{ed}$) and knowledge transfer loss ($\mathcal{L}_{kt}$), that are both computed solely on current-task data. Indeed, as shown in Figure 4, we empirically observed a slight performance drop when SAM was applied to all losses indiscriminately, further justifying our selective design choice.

Specifically, to ensure a flat minimum for these two losses, we modify each objective by incorporating worst-case perturbations within a neighborhood of the model parameters, following Equation (29).

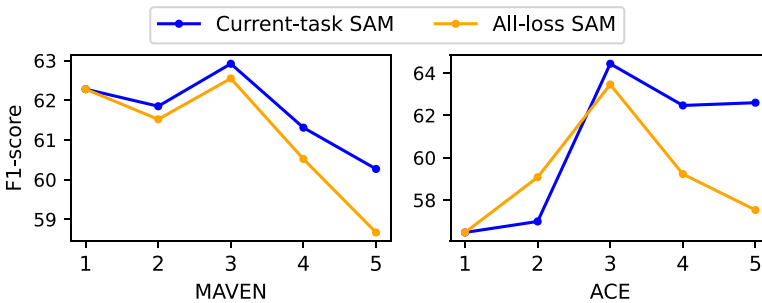


Figure 4

Comparison of applying SAM only to current-task losses versus all losses on MAVEN and ACE datasets.

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}^i\|_2 \leq \rho} \mathcal{L}_i(\boldsymbol{\theta} + \boldsymbol{\epsilon}^i), \quad (29)$$

Consequently, based on Equations (27) and (28), the gradient of the modified loss functions ($\mathcal{L}_{ed}$ and $\mathcal{L}_{kt}$) with respect to the model parameters, $\mathbf{g}_{ed}^{SAM}$ and $\mathbf{g}_{kt}^{SAM}$, are obtained as follows:

$$\begin{aligned} \boldsymbol{\epsilon}_{ed}^* &= \rho \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{ed}(\boldsymbol{\theta})}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{ed}(\boldsymbol{\theta})\|_2}, & \boldsymbol{\epsilon}_{kt}^* &= \rho \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{kt}(\boldsymbol{\theta})}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{kt}(\boldsymbol{\theta})\|_2}, \\ \mathbf{g}_{ed}^{SAM} &= \nabla_{\boldsymbol{\theta}} \mathcal{L}_{ed}(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{ed}^*), & \mathbf{g}_{kt}^{SAM} &= \nabla_{\boldsymbol{\theta}} \mathcal{L}_{kt}(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{kt}^*) \end{aligned} \quad (30)$$

As a result, we obtain gradient values for $\mathcal{L}_r$, $\mathcal{L}_d$, $\mathcal{L}_{ed}$, $\mathcal{L}_{kt}$, $\mathcal{L}_{LMH}$, and $\mathcal{L}_{reg}$ as $\mathbf{g}_r$, $\mathbf{g}_d$, $\mathbf{g}_{ed}^{SAM}$, $\mathbf{g}_{kt}^{SAM}$, $\mathbf{g}_{LMH}$, and $\mathbf{g}_{reg}$. Consequently, any gradient-based MOO technique can be readily applied. For simplicity, we omit the SAM notation and define the gradient set at training step τ as $\mathcal{G} = \{\mathbf{g}_i^{(\tau)}\}_{i \in \{r, d, ed, kt, LHM, reg\}}$. Below, we illustrate its application with NashMTL and ExcessMTL, considering the optimization step τ .

- **NashMTL:** Let $G^{(\tau)}$ represent the matrix whose columns are the gradients $\mathbf{g}_i^{(\tau)}$ and let $w^{(\tau)}$ represent the corresponding weight matrix for each gradient, where $i \in \{r, d, ed, kt, LHM, reg\}$. Similar to Equation (9), the weight matrix used to aggregate the gradients is determined by solving the following equation:

$$G^{(\tau)\top} G^{(\tau)} w^{(\tau)} = 1/w^{(\tau)}$$

- **ExcessMTL:** We first calculate the Hessian matrix $H_i^{(\tau)} \approx \text{diag}(\sum_{j=1}^{\tau} \mathbf{g}_i^{(j)} \mathbf{g}_i^{(j)\top})^{1/2}$. Consequently, the excess risks and gradient aggregation weights are calculated as in Equation (11):

$$\mathcal{E} = \mathbf{g}_i^{(\tau)\top} H_i^{(\tau)-1} \mathbf{g}_i^{(\tau)}; \quad w_i^{(\tau)} = w_i^{(\tau-1)} \exp(\eta_w \mathcal{E})$$

Finally, model parameters $\boldsymbol{\theta}$ are updated as:

$$\boldsymbol{\theta}^{(\tau)} = \boldsymbol{\theta}^{(\tau-1)} - \eta \sum_{i \in \{r, d, ed, kt, LHM, reg\}} w_i^{(\tau)} \mathbf{g}_i^{(\tau)} \quad (31)$$

where η represents the learning rate. The overall process of SaMOO Sequential Learning for CED is outlined in Algorithm 2.

3.4 Data Balancing via Synthetic Representations

A key challenge in continual event detection lies in the constrained size of the replay buffer, which becomes increasingly limiting as the dataset expands over the model's lifespan. This constraint introduces two major issues: the potential for overfitting to the

Algorithm 2. SaMOO Sequential Learning for Continual Event Detection**Input:**

Model parameters θ
 Perturbation radius ρ , learning rate η
 MOO aggregation algorithm $\mathcal{A}$
 Differentiable loss functions $\mathcal{L}_r, \mathcal{L}_d, \mathcal{L}_{LMH}, \mathcal{L}_{reg}, \mathcal{L}_{ed}$, and $\mathcal{L}_{kt}$.

Output:

Updated parameter θ^*

Optimization Procedure:

for $i \in [r, d, LMH, reg, ed, kt]$ **do**

$\mathbf{g}_i^{\text{loss}} = \nabla_{\theta} \mathcal{L}_i(\theta)$ ▷ Compute gradient
if $i \in \{ed, kt\}$ **then**
 $\mathbf{e}^{*i} = \rho \cdot \mathbf{g}_i^{\text{loss}} / \|\mathbf{g}_i^{\text{loss}}\|$ ▷ Compute worst-case perturbation
 $\mathbf{g}_i = \nabla_{\theta} \mathcal{L}_i(\theta + \mathbf{e}^{*i})$ ▷ Approximate SAM's gradient
else
 $\mathbf{g}_i \leftarrow \mathbf{g}_i^{\text{loss}}$

Calculate gradient aggregation weight vector w :

$$w = \mathcal{A}(\mathbf{g}_r, \mathbf{g}_d, \mathbf{g}_{LMH}, \mathbf{g}_{reg}, \mathbf{g}_{ed}, \mathbf{g}_{kt})$$

Update model parameter:

$$\theta^* = \theta - \eta \sum_{i \in \{r, d, LMH, reg, ed, kt\}} w_i \mathbf{g}_i$$

stored samples and data imbalance when using a multi-task replay strategy, both of which can degrade the effectiveness of rehearsal over time. To address these challenges and improve the diversity of stored representations, we introduce a Synthetic Representation (SR) module that leverages generative models, such as Gaussian Mixture Models (GMMs) (Reynolds et al. 2009), Variational Autoencoders (VAEs) (Kingma and Welling 2014), and Conditional Variational Autoencoders (cVAEs) (Sohn, Lee, and Yan 2015), to generate event-type representations. This approach not only enhances the robustness of memory-based methods but also helps balance the dataset, thereby supporting the effectiveness of the MOO strategy and improving the representational capacity of the LMH preservation mechanism.

It is important to highlight that generating high-quality natural language samples for event detection is particularly challenging due to the discrete nature of the data. To overcome this, we propose utilizing a frozen BERT model to learn the distribution of latent trigger representations, which encapsulate class-level characteristics. This approach is more practical than directly generating textual data, as it circumvents challenges related to grammar, sentence structure, and other linguistic complexities. Additionally, it provides substantial advantages in effectively classifying these representations.

After completing training on task t , a generative model (GM) is trained for each event type c in C_t , representing all event types at time t , to learn the latent

representations of the corresponding data. The trained GM is then stored for future sampling. When transitioning to the next task ($t + 1$), for each event type $c \in C_t$, the corresponding GM is used to generate $\tilde{n}$ synthetic trigger representations. The generated samples form the set $\tilde{R}$. The event features of instances in the replay buffer R are then combined with $\tilde{R}$ to create an augmented set, R_a . This new set, R_a , replaces R in the experience replay and distillation tasks (Yu, Ji, and Natarajan 2021). Consequently, the replay loss (L_r), defined as the cross-entropy loss over the augmented replay set R_a , and the distillation loss (L_d) can be formulated as follows:

$$L_r = -\frac{1}{m_{R_a}} \sum_{z_a \in R_a} \log(p_{z_a}) \quad (32)$$

$$L_d = -\frac{1}{m_{R_a}} \sum_{z_a \in R_a} p_{z_a}^{t-1} \log(p_{z_a}^t) \quad (33)$$

In the above equations, m_{R_a} denotes the number of trigger representations in R_a , and $p_{z_a} = \text{Softmax}(\text{MLP}(z_a))$.

4. Experiments

In this section, we present empirical results and analysis to demonstrate the effectiveness of our contributions in improving CED, in comparison against several state-of-the-art baselines.

4.1 Experimental Settings

4.1.1 Datasets. We conduct evaluation on four datasets: ACE (Walker et al. 2006) and MAVEN (Wang et al. 2020), which are standard benchmarks in CED, as well as two additional datasets, RAMS (Ebner et al. 2020) and GENEVA (Parekh et al. 2023). All of them are preprocessed following Yu, Ji, and Natarajan (2021). Detailed statistics of all datasets are provided in Tables 1 and 2.

Standard Benchmarks for CED. Following common practice in the CED literature (Yu, Ji, and Natarajan 2021; Wang, Wang, and Hu 2023; Liu, Chang, and Huang 2022; Qin et al. 2024), we primarily evaluate our method on ACE (Walker et al. 2006) and MAVEN (Wang et al. 2020), which are widely adopted benchmarks for studying event detection under incremental and continual learning settings. ACE provides a well-established

Table 1

Statistics of four datasets. #Doc stands for the total number of documents.

	#Doc	#Sentence	#Mention	#Negative	#Doc	#Sentence	#Mention	#Negative
	MAVEN				RAMS			
Train	2,522	27,983	67,637	280,151	7,327	7,327	7,327	28,886
Dev	414	4,432	10,880	46,318	922	922	922	3,509
Test	710	8,038	18,904	79,699	871	871	871	3,420
	ACE				GENEVA			
Train	501	18,246	4,088	261,027	1,968	1,968	4,170	5,038
Dev	41	1,846	433	53,620	783	783	1,442	1,906
Test	55	689	790	93,159	933	933	1,893	2,442

Table 2
Statistics of task subsets for four datasets. Each subset simulates a task in the continual event detection sequence.

Subset	MAVEN		ACE		RAMS		GENEVA	
	#Type	#Mention	#Type	#Mention	#Type	#Mention	#Type	#Mention
1	33	12,927	9	584	26	1,187	25	859
2	30	12,292	6	839	30	1,782	22	856
3	39	14,438	5	1,334	31	1,779	26	860
4	35	13,413	5	719	31	1,345	25	798
5	31	14,567	8	612	21	1,243	17	797

ontology with clearly defined event types and has been extensively used in prior CED work, making it a standard reference point for evaluating catastrophic forgetting and backward transfer. **MAVEN**, in contrast, offers a substantially larger and more diverse event ontology, with 168 event types and fine-grained semantic distinctions. Its scale and semantic richness make MAVEN particularly suitable for assessing a model’s ability to continually acquire new event knowledge while preserving previously learned representations. Together, ACE and MAVEN enable fair comparison with prior work while covering both moderate-scale and large-scale CED scenarios.

Additional Datasets for Broader CED Evaluation. To further strengthen the empirical evaluation and address concerns about dataset diversity, we have expanded our evaluation to include two additional publicly accessible datasets, RAMS (Ebner et al. 2020) and GENEVA (Parekh et al. 2023), both of which contain rich and diverse event structures. Specifically, RAMS is a large-scale English dataset for event argument extraction. It contains over 9,000 documents annotated with 139 event types, where each instance focuses on a single event trigger and its corresponding arguments within a document. **RAMS** is designed to emphasize argument role classification under realistic settings, with long documents and diverse event schemas. Additionally it contains long, information-dense sentences with highly fine-grained event categories, leading to increased semantic overlap and inherent ambiguity between event types. This setting amplifies both catastrophic forgetting and semantic interference, making it particularly challenging for continual learning. **GEVENA** is a benchmark event detection and event argument extraction dataset designed to evaluate how well NLP models generalize to diverse and unseen events. It contains a large ontology of 115 event types and 220 argument roles with expert annotations and multiple test suites to challenge models on identifying event triggers, classifying event types, and extracting associated arguments beyond traditional limited schemas. These additions allow us to validate the generality of our findings across datasets with different characteristics.

Discussion on Available ED Datasets. There exist many ED datasets in the literature (Song et al. 2015; Veyseh et al. 2022; Parekh et al. 2024); however, not all of them are suitable for constructing realistic CED scenarios, which require a sufficiently rich ontology, a large number of event types, and the ability to incrementally introduce new semantic categories over time. For instance, Rich ERE (Song et al. 2015) is built on top of Light ERE, which was explicitly designed as a simplified version of the ACE ontology. As a result, Rich ERE contains only 9 top-level event types with 38 subtypes, with an event schema only marginally larger than ACE, making its domain coverage

and event schema highly similar to ACE and offering limited additional challenges for continual learning beyond what is already observed on ACE. Similarly, the MEE dataset (Veyseh et al. 2022) is valuable for multilingual evaluation, but it strictly adopts the ACE event ontology, with only 8 event types and 16 subtypes, identical to the original ACE inventory. Although multilinguality is orthogonal and interesting, MEE does not expand the event ontology or increase the number of event categories, and therefore does not introduce a more demanding CED setting in terms of incremental semantic expansion. In contrast, SPEED (Parekh et al. 2024) focuses on a narrow domain with only seven epidemic-related event types (e.g., *Infection*, *Recovery*), which is insufficient to support a realistic CED setup where models must continually adapt to diverse and evolving event semantics. Overall, these datasets contain fewer than 40 event types (Rich ERE: 38, MEE: 16, SPEED: 7), whereas MAVEN, RAMS, and GENEVA comprise over 100 event types, providing substantially richer ontologies that are better suited for studying scalability, catastrophic forgetting, and semantic interference in CED.

4.1.2 Baselines. In this section, we describe the baseline methods used in our experiments. Table 3 provides a structured categorization of these baselines according to their underlying continual learning paradigms, including memory-based, regularization-based, and parameter-efficient fine-tuning approaches. In summary, our baseline selection covers a broad and representative spectrum of continual learning strategies, ranging from generic CL methods adapted to the CED setting to state-of-the-art approaches explicitly tailored for CED.

CED Baselines. Within the CED literature, we consider five representative state-of-the-art methods as baselines: Knowledge Consolidation Networks (KCN) (Cao et al. 2020), Knowledge Transfer (KT) (Yu, Ji, and Natarajan 2021), Episodic Memory Prompt (EMP) (Liu, Chang, and Huang 2022), Embedding Space Separation and Compaction (ESCO) (Qin et al. 2024), and Semantic Confusion Rectification (SCR) (Wang, Wang, and Hu 2023). Although these methods are specifically proposed for CED, they fundamentally build upon core continual learning paradigms, including memory replay, knowledge distillation, regularization, and parameter-efficient fine-tuning (PEFT). Specifically, **KCN** (Cao et al. 2020) uses a limited memory buffer for replay, combined with knowledge distillation and prototype-enhanced retrospection to alleviate catastrophic forgetting. **KT** (Yu, Ji, and Natarajan 2021) adopts a memory-based framework

Table 3
Classification of CED methods by their primary continual learning mechanisms. “Problem” denotes the evaluation task used in the original work.

Method	Problem	Memory	Regularization	PEFT
iCaRL (Rebuffi et al. 2017)	Image Classification	✓	✓	✗
EEIL (Castro et al. 2018)	Image Classification	✓	✓	✗
BIC (Wu et al. 2019)	Image Classification	✓	✓	✗
KCN (Cao et al. 2020)	Event detection	✓	✓	✗
KT (Yu, Ji, and Natarajan 2021)	Event detection	✓	✓	✗
SCR (Wang, Wang, and Hu 2023)	Event detection	✓	✓	✗
EMP (Liu, Chang, and Huang 2022)	Event detection	✓	✓	✓
ESCO (Qin et al. 2024)	Event detection	✓	✓	✓
MOZILA (Our)	Event detection	✓	✓	✗

that integrates knowledge distillation and knowledge transfer, where samples of new event types reinforce previously learned knowledge, while samples of old event types help initialize representations for newly introduced classes. **EMP** (Liu, Chang, and Huang 2022) further extends knowledge distillation by introducing soft prompts into the input text, enabling parameter-efficient preservation of prior knowledge. **ESCO** (Qin et al. 2024) builds upon EMP and combines embedding space separation and compaction by pushing new data away from old data to reduce interference, while pulling replayed samples toward their class prototypes to enhance intra-class compactness and mitigate overfitting. **SCR** (Wang, Wang, and Hu 2023) addresses semantic ambiguity in CED by incorporating pseudo-labeling, knowledge transfer, and explicit strategies to strengthen the learning of long-tailed event types.

Standard CL Baselines. In addition to CED-specific methods, we also include several standard continual learning approaches that are directly adapted to the event detection setting. We consider a simple **Fine-tuned** baseline, which sequentially trains the model on each task and is known to be highly susceptible to catastrophic forgetting. Incremental classifier and representation learning (**iCaRL**) (Rebuffi et al. 2017) leverages an exemplar memory and knowledge distillation for class-incremental learning. End-to-end incremental learning (**EEIL**) (Castro et al. 2018) selects representative samples from new classes and jointly trains them with replayed data to mitigate data imbalance. Bias correction (**BIC**) (Wu et al. 2019) applies an affine transformation to correct the bias toward newly introduced classes. We also include an **Upperbound** model trained jointly on all tasks to estimate ideal performance. The implementations of iCaRL, EEIL, BIC, and KCN are adopted from the codebase released by Yu, Ji, and Natarajan (2021).

We evaluate all baseline methods on the standard datasets ACE and MAVEN. For the additional datasets RAMS and ESCO, due to the high computational cost of continual training across multiple tasks and runs, we compare MOZILA against the top-five performing and representative baselines; selected based on their performance on ACE and MAVEN, namely *BIC*, *KT*, *SCR*, *EMP*, and *ESCO*. To ensure a fair comparison, all baseline methods are re-executed on the same preprocessed datasets.

4.1.3 Experimental Configuration.

Experimental Scenario. To conduct our experiment, we adopt the Oracle negative setting (Yu, Ji, and Natarajan 2021). Specifically, to simulate continual event detection, both MAVEN and ACE datasets are partitioned into five subsets, each corresponding to a sequential task. As shown in Table 2, each subset introduces a distinct set of event types and their corresponding mentions. Under this setting, at each stage t , the model is trained on event types and instances from subset D_t , while being evaluated on all types seen so far from $D_{1:t}$. This setup defines a sequence of five continual learning tasks with an expanding event ontology. A special label NA (i.e., “no event”) is included in all tasks. It accounts for two kinds of examples: (1) negative samples from the full dataset (Table 1), and (2) mentions of unseen types (i.e., those from future tasks $D_{t+1:5}$) that are treated as negative during training. During training, we use the AdamW optimizer (Loshchilov and Hutter 2019) with a learning rate of $1e^{-4}$ and a weight decay of $1e^{-2}$. For both datasets, the batch size is set to 128, and early stopping is applied, terminating the training process after 5 epochs if no improvement is observed on the development set. For the KT baseline (Yu, Ji, and Natarajan 2021), a replay set is maintained with 20 instances per label, and 20 instances are used for generating representations of each new

label. For each method, the best results are obtained through a grid search. The search ranges for each hyperparameter are as follows:

- the generation ratio r is selected from $\{2, 5, 10, 20\}$.
- the number of training epochs is chosen from 15 or 30.
- the balancing coefficient v , which controls the ratio between NA and valid labels, is selected from $\{\frac{4}{5}, \frac{10}{11}, \frac{20}{21}, \frac{30}{31}, \frac{40}{41}\}$.
- the number of components in GMMs, denoted as g , is selected from $\{2, 4, 6, 8\}$. Additionally, for event types that have more than 600 instances, we double the value of g in our experiments.

Permutation Setup. To ensure a robust evaluation of continual learning models, we adopt a *permutation-based task ordering strategy* following Yu, Ji, and Natarajan (2021). Specifically, the order in which task subsets are presented during training is controlled by a set of predefined permutations. This setup is important because the order of task arrival can significantly influence the degree of knowledge retention and forgetting in continual learning scenarios. We define five permutations over the five task subsets, as shown below:

$$\text{PERM} = \{[1, 2, 3, 4, 5], [5, 4, 3, 2, 1], [1, 4, 2, 5, 3], [2, 3, 1, 4, 5], [4, 5, 1, 2, 3]\}$$

These permutations are uniformly applied across all evaluated methods to ensure a fair and consistent comparison.

Evaluation Metrics. We use the task permutations and report the average F1 score for each task across five different permutations. Specifically, we use the F1 score to measure the model’s performance across tasks. In conventional lifelong learning evaluations for task $\mathcal{T}_i$, the model is assessed only on instances corresponding to previously learned types. However, in event detection, the model is continuously evaluated on the entire test set, treating mentions of unlearned event types as negative instances.

Infrastructure. All implementations are developed using PyTorch (Paszke et al. 2019), and all experiments were conducted on an NVIDIA A100 GPU and an NVIDIA V100 GPU, each with 80GB of VRAM.

4.2 Comprehensive Evaluation of MOZILA

In this section, we conduct a comprehensive analysis of the key components in MOZILA, namely, MOO and LMH preservation. We also highlight the superior performance of MOZILA in comparison with baseline methods. We present the model’s performance using the two best-performing techniques: NashMTL for MOO, OT for LMH preservation, and GMM for Synthetic Representations. Additional results using alternative methods are discussed in Section 4.3.

Impact of Learning with Language Model Head Preservation. From Figure 5a, it is evident that incorporating linguistic information from the LMH leads to clear performance improvements. Specifically, applying OT for alignment yields performance

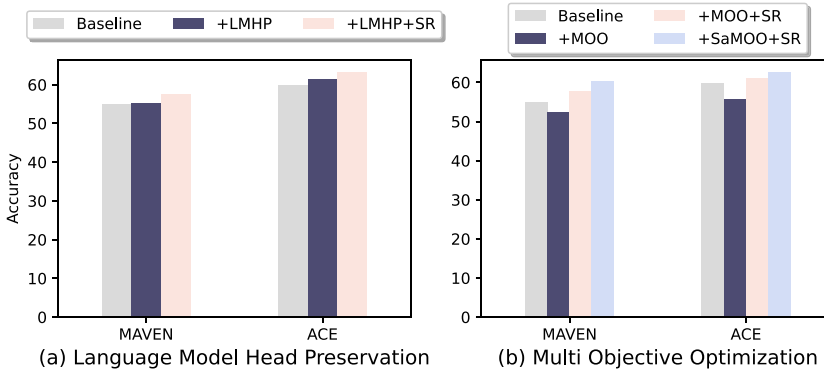


Figure 5

The final task accuracy after training using (a) LMHP with OT and (b) MOO with NashMTL on the Baseline model for the MAVEN and ACE datasets. In this context, the Baseline corresponds to the KT method, while SR represents Synthetic Representations, and SAM stands for Sharpness-Aware Minimization.

gains of 0.34% and 1.65% on the respective benchmarks. Furthermore, the integration of Synthetic Representations leads to an additional improvement of around 3% on both datasets, indicating that this strategy effectively supports LMHP preservation and further enhances the model’s representation capability.

Impact of Sharpness-Aware Multi-Task Learning on CED. We investigate the use of MOO to address multiple learning objectives within the CED framework. In particular, we report the performance of integrating NashMTL, a gradient-based MOO method, into the KT baseline. Figures 3 and 5b illustrate that directly incorporating MOO techniques into CED without proper adaptation results in degraded performance. As discussed in Section 1, we hypothesize that one of the primary challenges stems from the severe class imbalance in later tasks, where current-task classes dominate the training data, while earlier-task classes are underrepresented in the replay buffer. Indeed, Figure 5b demonstrates that incorporating SRs to address data imbalance effectively stabilizes the application of MOO and leads to substantial improvements over the baseline in the CED setting. Furthermore, our carefully designed framework, which integrates SAM specifically adapted for continual learning, effectively identifies minimizers in flat regions of the loss landscape. When combined with MOO techniques, this leads to superior performance compared with baseline.

Pushing the State of the Art in CED with MOZILA. Previous discussions have demonstrated the individual effectiveness of both LMHP and SaMOO for CED. LMHP enhances the model’s ability to retain essential linguistic knowledge, improving representation quality and generalization, while SaMOO, through SAM and MOO tailored for continual learning, enables the model to converge to flatter minima that are more robust to task shifts. Additionally, Tables 4 and 5 demonstrate that each component, when applied independently, consistently surpasses all baseline methods across both benchmarks, highlighting their individual contributions to the overall effectiveness of the framework. For instance, incorporating only LMHP and SaMOO surpasses ESCO by approximately 0.2% and 3% on MAVEN, respectively. Similarly, on the ACE dataset, both LMHP and SaMOO individually outperform the strongest baseline by

Table 4
Classification F1-scores (%) on two datasets MAVEN and ACE. **wo X** indicates that module X is removed from the MOZILA framework during evaluation.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
Fine-tuned	63.16	40.30	33.00	23.86	22.34	55.88	40.25	37.04	20.71	25.01
iCaRL	18.08	27.03	30.78	31.26	29.77	5.05	6.42	7.05	6.93	9.44
EEIL	63.16	48.17	44.17	40.35	37.75	55.88	48.63	57.14	50.45	52.68
BIC	63.16	55.51	53.96	50.13	49.07	55.88	58.16	61.23	59.72	59.02
KCN	63.16	55.73	53.69	48.86	47.44	55.88	58.55	61.40	59.48	58.64
KT	62.76	58.49	57.46	55.38	54.87	55.88	57.29	61.42	60.78	59.82
EMP	66.82	58.02	58.19	55.07	54.52	59.05	57.14	55.80	53.42	52.97
ESCO	67.66	60.25	59.56	56.50	56.06	55.20	55.86	56.99	54.57	55.01
SCR	76.52	57.97	57.89	52.74	53.41	75.24	63.3	61.07	55.05	55.37
MOZILA	63.30	61.97	63.00	61.81	61.49	60.15	59.73	64.55	63.65	64.27
wo LMHP	62.28	61.85	62.92	61.31	60.27	56.47	56.99	64.44	62.47	62.60
wo SaMOO	62.98	60.47	60.78	58.53	57.53	58.30	59.69	63.52	61.05	63.22
Upperbound	/	/	/	/	63.46	/	/	/	/	67.95

Table 5
Classification F1-scores (%) on two datasets RAMS and GENEVA. **wo X** indicates that module X is removed from the MOZILA framework during evaluation.

Task	RAMS					GENEVA				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
BIC	38.30	39.49	38.48	36.14	33.28	82.73	80.91	77.63	74.47	70.13
KT	39.90	40.41	37.64	37.86	36.33	79.76	78.06	75.70	75.20	71.75
EMP	46.26	44.76	44.76	42.08	39.61	85.63	71.37	78.44	79.48	77.71
ESCO	49.79	46.74	43.08	40.37	41.11	85.87	76.93	77.12	78.58	79.09
SCR	69.00	52.06	44.02	38.09	34.24	85.74	80.56	77.72	76.27	75.78
MOZILA	49.40	46.08	45.23	43.31	42.69	81.39	82.31	82.48	83.12	82.94
wo LMHP	49.42	46.10	45.10	43.14	42.37	79.61	78.65	79.20	80.71	81.10
wo SaMOO	47.92	46.31	45.60	45.59	41.87	81.39	81.82	80.38	80.33	78.91

approximately 3–4% at the end of the training sequence. Additionally, among the two additional datasets, Table 5 shows that RAMS represents the most challenging setting. It contains *long and information-dense sentences*, *long-tail distribution*, and *fine-grained categories* across event types. These properties exacerbate both catastrophic forgetting and gradient dominance by frequent classes. In contrast, GENEVA yields higher absolute performance for most methods. This is largely because the NA class is less dominant, which alleviates extreme class imbalance and reduces the difficulty of continual adaptation. Despite this, when integrated within MOZILA, these components complement and reinforce one another, resulting in a synergistic effect that elevates the performance to the highest observed levels, with F1 scores reaching 61.49% on MAVEN, 64.27% on ACE, 42.69% on RAMS, and 82.94% on GENEVA. These findings affirm the efficacy and robustness of MOZILA, establishing it as a leading state-of-the-art framework for CED systems.

To further assess the robustness of MOZILA across different runs, we conducted formal statistical testing using a two-sided Wilcoxon signed-rank test (Wilcoxon 1992; Ghose and Nguyen 2024) across five runs per task, starting from Task 2, and report

Table 6
Wilcoxon signed-rank test (W) and Cliff’s delta (δ) comparing MOZILA against KT and ESCO across Tasks 2–5 on RAMS, GENEVA, ACE, and MAVEN.

Task	RAMS		GENEVA		ACE		MAVEN	
	W	δ	W	δ	W	δ	W	δ
MOZILA vs. KT								
2	0.0625	1.00	0.0625	1.00	0.8125	−0.04	0.0625	0.84
3	0.0625	1.00	0.0625	1.00	0.4375	0.04	0.0625	1.00
4	0.0625	0.84	0.0625	1.00	0.0625	0.60	0.0625	1.00
5	0.0625	1.00	0.0625	1.00	0.0625	0.52	0.0625	1.00
MOZILA vs. ESCO								
2	0.8125	0.04	0.1250	0.84	0.0625	0.68	1.00	0.36
3	0.1250	0.76	0.0625	1.00	0.0625	0.92	0.0625	0.76
4	0.0625	0.68	0.0625	1.00	0.0625	1.00	0.0625	1.00
5	0.1250	0.60	0.0625	1.00	0.0625	1.00	0.0625	1.00

Cliff’s delta (Cliff 1993) to quantify effect size (Table 6). Due to the small number of runs ($n = 5$), the Wilcoxon signed-rank test commonly yields a p-value of 0.0625, which is the smallest attainable value under this setting, consistently showing borderline-significant Wilcoxon statistics paired with large to very large effect sizes (Cliff’s $\delta \geq 0.6$) across all datasets. Importantly, effect sizes increase in later tasks, particularly on ACE, indicating that our method maintains stronger performance as more tasks are introduced. In addition, we provide a complementary analysis of *Forgetting and Backward transfer* (Figure 6), where we inspect task-wise performance trajectories over time. These analyses show that MOZILA achieves more stable performance and consistently positive backward

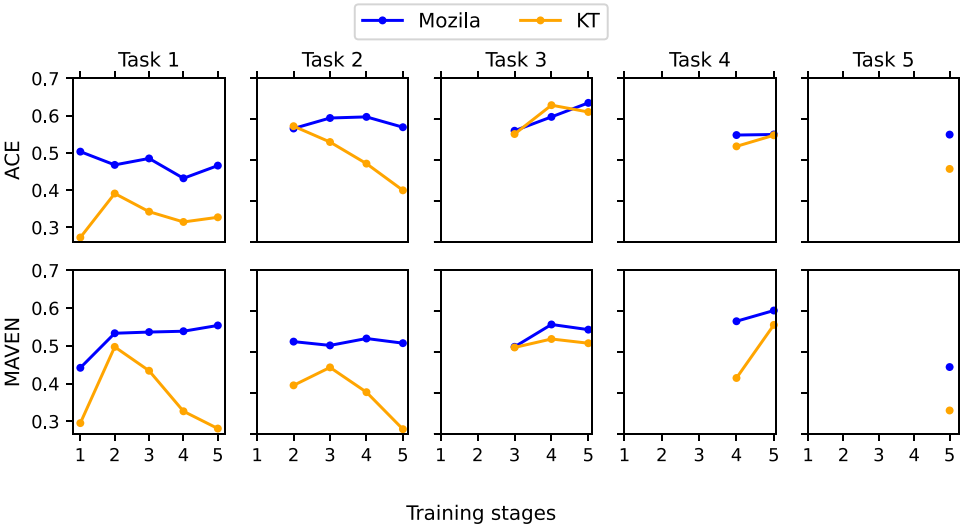


Figure 6
Macro-F1 scores of event types first introduced at each task, tracked across subsequent training stages. The results highlight the degree of forgetting after the initial learning of new event types and their retention over time as additional tasks are introduced. Macro-F1 is used to reduce the dominance of event types with larger training data.

transfer compared with baseline, further supporting its ability to mitigate forgetting while effectively leveraging knowledge from later tasks. Therefore, these results provide empirical evidence for improved robustness against forgetting and better adaptation to new event types, rather than a speculative interpretation of point-wise gains.

Forgetting and Backward Transfer. From the results in Table 4, we can observe the overall performance of models on the extended event ontology, including the final F1 scores when evaluated on all event types. However, these aggregate results alone do not reveal the extent of forgetting that occurs during sequential learning. To further investigate the phenomena of catastrophic forgetting and backward transfer, we analyze the performance of event types after their initial introduction and track their F1 scores across subsequent training stages. Specifically, we report macro-F1 scores to reduce the dominance of event types with large training data, allowing us to better capture the retention of less frequent types. As shown in Figure 6, both ACE and MAVEN datasets exhibit clear forgetting trends: for KT, the performance of early-introduced types (e.g., Task 1 and Task 2) declines sharply as new tasks are learned, highlighting its vulnerability to catastrophic forgetting. In contrast, MOZILA consistently maintains higher macro-F1 across stages, demonstrating its stronger ability to preserve knowledge of earlier tasks. Moreover, we observe backward transfer in later tasks, where the introduction of new data can sometimes benefit performance on earlier types (Tasks 3 and 4). For instance, both methods show modest performance gains on certain tasks after subsequent training; however, MOZILA achieves more stable and positive backward transfer compared to KT. These results underscore the robustness of MOZILA in both mitigating forgetting and leveraging backward transfer to improve CED.

Computational Overhead. Table 7 shows that while the baseline is the most efficient, it yields much lower performance. Our proposed components LMHP (+OT) and SaMOO add moderate overhead ($3.3\times$ and $1.9\times$ on MAVEN). This rise in cost stems from the added optimization steps (e.g., distribution alignment in OT and multi-objective balancing in SaMOO), which bring moderate but reasonable accuracy gains. MOZILA incurs the highest training cost, yet delivers over 6% F1 improvement, making the trade-off worthwhile. Finally, it is worth noting that the inference stage remains unaffected across all methods. Once trained, the models have comparable runtime efficiency during prediction, meaning the additional cost is confined only to the training phase.

Table 7
Training Time Complexity ($\downarrow$) induced by different alignment strategies and proposed components (seconds per epoch). Measurements are conducted on an NVIDIA A100 GPU with 80GB of VRAM.

	MAVEN	ACE
Baseline	58.51	21.22
+KL	68.58 $\times 1.17$	28.78 $\times 1.36$
+OT	195.67 $\times 3.34$	66.63 $\times 3.14$
+MIM	320.17 $\times 5.47$	126.58 $\times 5.97$
+SaMOO	109.21 $\times 1.87$	36.44 $\times 1.72$
MOZILA	338.86 $\times 5.79$	162.85 $\times 7.67$

Key Takeaways

- *Effectiveness of Proposed Components:* Both LMHP and SaMOO individually improve performance over all baselines, confirming their robustness and complementary strengths in addressing representation quality and multi-task objectives.
- *State-of-the-Art Framework with MOZILA:* When integrated, these two components reinforce each other within the MOZILA framework, achieving the highest F1 scores of 61.49% on MAVEN and 64.27% on ACE, establishing MOZILA as a new state-of-the-art for CED.

4.3 Ablation Study

In this section, we investigate the impact of various generative models and MOO methods on the overall performance. In addition, we report results for different distribution alignment strategies used in LMHP, including KL divergence, MIM, and OT. We further provide ablation studies on the ratio of synthesized representations and the number of GMM components in the SR module. Lastly, we examine the sensitivity of key hyperparameters associated with each component to assess the robustness of the proposed framework. In our ablation studies, we adopt KT as the baseline, given its strength and wide use in CED as well as its role as the foundation for many follow-up works. Because our framework extends KT with additional components, it provides the most suitable reference point to assess their contributions. Given the substantial computational cost of running ablations under extensive continual settings, we limit these experiments to ACE and MAVEN, which are standard benchmarks in CED.

4.3.1 Systematic Analysis of the LMH Preservation. In this section, we explore various alignment strategies for LMHP, examine the impact of the temperature hyperparameter τ in Equation (15), and analyze the effect of the balancing coefficient γ (Equation 25) used for the Class Embedding Regularization.

Exploring Alignment Methods for Preserving LMH. As outlined in Section 3.2, we introduced three approaches for preserving the LMH: KL divergence, MIM, and OT. Figure 7 demonstrates that all methods consistently enhance model performance, substantiating the effectiveness of our framework in preserving and utilizing information from the LMH. Specifically, OT and MIM achieve competitive performance, with OT yielding the best results on MAVEN, while MIM outperforms other methods on ACE at task $\mathcal{T}_5$. Their capacity to capture fine-grained discrepancies across distributions makes it particularly well-suited for retaining linguistic information in a continual learning setting. However, OT's and MIM's computational complexity can be a limiting factor in practice. In such cases, KL divergence offers a computationally efficient alternative (see Table 7). Despite its simplicity, KL still yields significant improvements and consistently outperforms

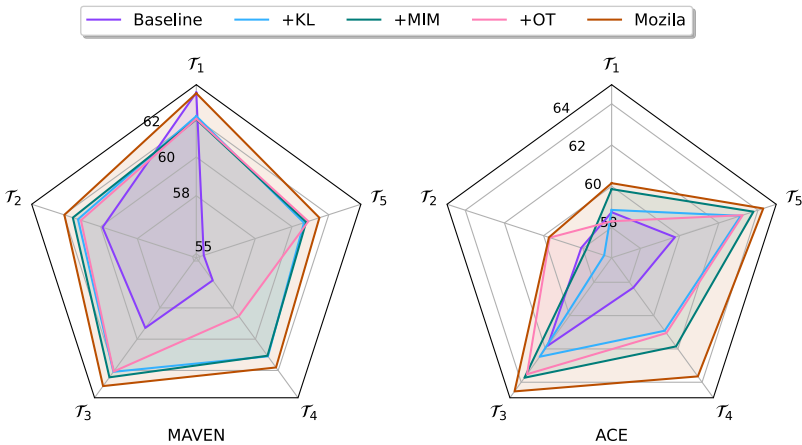


Figure 7
F1 score after training each task using different alignment methods applied to the Baseline. In this context, Baseline denotes the MOZILA framework with both the LMHP and SaMOO components excluded.

the baseline, making it a practical and effective solution when resource constraints are present. To balance between performance and time complexity, we opt to use OT in MOZILA, which further achieves superior performance after each task, consistently encompassing the entire range of F1 scores attained by all baseline methods, as illustrated in Figure 7. These results underscore the flexibility of our LMH preservation module, which remains effective across various distribution alignment strategies and achieves further performance gains when integrated with the SaMOO component within the MOZILA framework.

Hyperparameter Sensitivity. Table 8 presents the ablation results for varying the temperature hyperparameter τ in the LMH alignment. Across both MAVEN and ACE datasets, the performance remains stable, with only minor fluctuations. For instance, the F1 scores on Task τ^5 vary within a narrow range—from 57.13% to 57.71% on MAVEN and from 61.15% to 63.22% on ACE, reflecting a maximum absolute difference of only 0.58% and less than 2%, respectively. Similarly, Table 9 presents the ablation results for varying the balancing coefficient γ used in the Class Embedding

Table 8
Ablation results for the temperature of the language modeling head.

Task	MAVEN					ACE				
	τ_1	τ_2	τ_3	τ_4	τ_5	τ_1	τ_2	τ_3	τ_4	τ_5
$\tau = 5$	63.15	60.78	60.66	58.51	57.37	57.41	59.00	63.60	60.87	61.81
$\tau = 4$	63.08	60.72	60.71	58.76	57.71	61.09	60.12	63.36	61.09	61.15
$\tau = 3$	63.06	60.77	60.70	58.43	57.30	58.09	59.46	63.98	61.63	62.36
$\tau = 2$	63.11	60.64	60.70	58.45	57.50	58.30	59.69	63.52	61.05	63.22
$\tau = 1$	62.98	60.47	60.78	58.53	57.53	60.42	59.76	64.28	61.52	62.84
$\tau = 0.1$	62.52	60.31	60.51	58.31	57.13	61.51	57.01	62.94	60.18	61.22
$\tau = 0.01$	62.60	60.3	60.43	58.22	57.15	62.15	57.08	63.51	59.48	61.29

Table 9
Ablation results for regularization on cross-task class representations.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
$\gamma = 1$	63.01	60.36	60.67	58.33	57.41	58.30	59.72	64.41	60.97	62.29
$\gamma = 0.5$	62.98	60.47	60.78	58.53	57.53	58.30	59.69	63.52	61.05	63.22
$\gamma = 0.2$	63.07	60.66	60.67	58.37	57.16	58.72	59.39	64.55	61.88	62.68
$\gamma = 0.1$	63.01	60.45	60.60	57.79	57.02	58.72	60.01	64.61	62.49	62.82

Regularization loss. Across both MAVEN and ACE benchmarks, the F1 scores exhibit high stability under varying γ values, with absolute differences of only 0.51% and 0.93%, respectively, indicating low sensitivity to this hyperparameter. These consistently high scores demonstrate the robustness of the LMH Preservation module and suggest that model performance is largely insensitive to the choice of hyperparameters, thereby reducing the computational cost typically associated with hyperparameter tuning.

Key Takeaways

- **Flexible and Effective Alignment Strategies for LMHP:** OT and MIM yields the comparative performance for LMH preservation, while KL divergence offers a computationally efficient alternative that still outperforms baselines.
- **Robustness to Hyperparameter Settings:** The LMHP module shows low sensitivity to both the temperature τ and balancing coefficient γ , reducing the need for extensive hyperparameter tuning.

4.3.2 Systematic Analysis of the Sharpness-Aware Multi-Objective Optimization. In this section, we assess SaMOO with various MOO strategies on CED, as outlined in Section 2.3, and highlight the effectiveness of our thoughtfully chosen design within the proposed framework. The results in Table 10 present the performance of various MOO methods, including both vanilla integrations and the design within MOZILA.

When MOO methods were used in isolation, FairGrad achieved the best performance across most tasks on both datasets, indicating its suitability for integration within the MOZILA framework. However, applying MOO techniques directly to the KT baseline without any adaptation led to performance degradation. For instance, FairGrad degraded KT’s performance by 4.16% on MAVEN and 0.62% on ACE, while NashMTL resulted in performance drops of 2.47% and 4.13% on MAVEN and ACE, respectively. The remaining MOO techniques exhibit even greater performance degradation. The main reasons for these decreases are the oversight of training data imbalance and the inherent differences between multi-task learning and continual learning. In MOZILA, the MOO technique is carefully integrated with synthetic representations to address data imbalance, while SAM is used to steer the optimization process toward flatter minima.

Table 10
Ablation results on MOO methods. **wo LMHP** indicates the removal of the LMHP module to evaluate the MOO techniques in isolation.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
KT	62.76	58.49	57.46	55.38	54.87	55.88	57.29	61.42	60.78	59.82
+FairGrad	63.69	60.10	58.67	54.12	50.71	55.19	57.97	62.87	61.14	59.20
+NashMTL	63.69	60.56	59.12	55.56	52.40	55.19	58.12	62.77	58.46	55.69
+ExcessMTL	63.08	59.66	56.65	50.59	47.81	55.38	57.99	61.68	58.49	56.42
+PCGrad	63.69	58.91	54.71	48.70	45.58	55.19	57.99	62.47	58.40	55.56
+IMTL	63.69	53.02	51.92	50.33	48.06	55.19	56.96	58.62	59.72	56.23
MOZILA wo LMHP										
$\mathcal{A}$ = FairGrad	62.28	61.71	63.04	61.45	60.52	56.67	58.60	63.83	63.26	63.31
$\mathcal{A}$ = NashMTL	62.28	61.85	62.92	61.31	60.27	56.47	56.99	64.44	62.47	62.60
$\mathcal{A}$ = ExcessMTL	62.28	62.35	62.52	59.51	58.15	56.64	57.74	64.53	61.46	61.41
$\mathcal{A}$ = PCGrad	62.28	61.45	61.52	57.78	55.48	56.47	58.36	63.62	61.53	61.58
$\mathcal{A}$ = IMTL	62.28	58.43	58.47	56.40	55.93	56.47	54.76	59.20	56.70	56.98

Experimental results demonstrate that this design substantially improves performance compared to standard MOO applications.

When incorporated into the MOZILA framework, FairGrad achieved the highest F1 scores across the majority of tasks, followed closely by NashMTL, consistently across both datasets. Specifically, on the MAVEN dataset, FairGrad achieved an F1 score of 60.52%, outperforming both NashMTL (60.27%) and PCGrad (55.48%). A similar trend is observed on the ACE dataset, where FairGrad delivered the highest performance, exceeding KT by 3.49%, while NashMTL achieved a 2.78% improvement. While FairGrad demonstrates strong performance, it requires additional computational overhead due to the need for tuning the α -fair hyperparameter, as defined in Equation (10), making it less

Key Takeaways

- **Effectiveness of SaMOO Design:** Naively applying MOO methods in continual learning settings leads to performance degradation, but when carefully integrated with synthetic representations and SAM, as in MOZILA, MOO techniques significantly improve performance—demonstrating the strength of the proposed SaMOO design.
- **Efficiency of Selected MOO Strategy:** Among the evaluated MOO methods, those integrated within the MOZILA framework show notable performance improvements. While FairGrad achieves the highest overall results, its additional tuning overhead makes NashMTL a more efficient and robust choice for practical deployment in MOZILA.

practical compared to NashMTL. Therefore, in our full implementation of MOZILA, we adopt NashMTL as the default MOO technique for performance comparison. Nonetheless, we believe that incorporating a more powerful MOO method like FairGrad has the potential to further enhance the effectiveness of MOZILA.

4.3.3 Systematic Analysis of the Synthetic Representation. As discussed in Section 4.2, SR plays a crucial role in enhancing the overall effectiveness of the MOZILA framework. In this section, we examine three generative models utilized for SR: GMMs (Reynolds et al. 2009), VAEs (Kingma and Welling 2014), and cVAEs (Sohn, Lee, and Yan 2015), with results indicating that GMMs are the most effective approach. We then further investigate two factors that may influence the performance of the SR module: the number of generated samples and the number of GMM components. Given the substantial influence of the SR module on SaMOO, we conduct our investigation in isolation with SaMOO, excluding LMHP to reduce the complexity and computational cost associated with hyperparameter tuning.

Impact of Generative Models in SR Module. The results presented in Table 11 show that all generative models contribute to performance gains over the baseline KT. When evaluated independently, GMMs yield the highest improvements, enhancing KT by 3.83% on MAVEN and 3.19% on ACE after the fifth task, underscoring the effectiveness of generative modeling in mitigating data imbalance. Furthermore, when integrated into the MOZILA framework with proposed MOO design, all generative approaches deliver additional performance gains across most tasks. GMMs remain the most effective, achieving an F1 score of 60.27% on MAVEN—1.57% higher than standalone GMMs; and also attaining the best result on ACE with a score of 62.60%. Additionally, GMMs offer greater efficiency in terms of training and storage. Unlike VAEs, which require a separate network per event type, or cVAEs, which demand one per task, GMMs avoid network proliferation, leading to reduced computational overhead during both training and inference. These findings underscore the importance of carefully selecting the generative model within the MOZILA framework.

To provide further analysis, we compare the quality of synthetic representations generated by GMMs and VAEs in Figure 8. For visualization, we selected four representative event types from each dataset: two frequent types with many samples

Table 11
Ablation results of generation methods in SR module.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
KT	62.76	58.49	57.46	55.38	54.87	55.88	57.29	61.42	60.78	59.82
+GMMs	62.76	59.30	59.55	58.25	57.70	55.88	56.97	62.48	60.88	62.01
+VAE	62.76	60.06	59.82	57.14	55.68	55.88	57.37	61.52	59.38	59.51
+cVAE	62.76	59.62	59.60	56.56	54.88	55.88	57.97	64.24	60.46	61.81
MOZILA wo LMHP										
SR = GMMs	62.28	61.85	62.92	61.31	60.27	56.47	56.99	64.44	62.47	62.60
SR = VAE	62.28	61.63	62.63	60.62	59.30	56.47	59.61	63.16	60.3	61.56
SR = cVAE	62.28	61.68	62.38	60.33	58.77	56.47	57.47	63.77	60.60	61.87

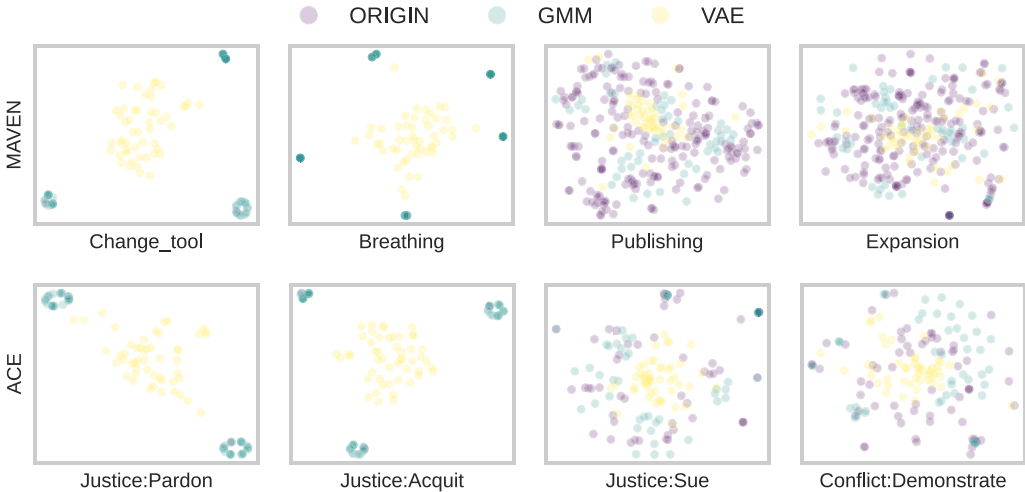


Figure 8
Visualization of synthetic trigger representations generated by GMM and VAE compared with original samples on MAVEN and ACE. Two rare and two frequent event types are shown for each dataset.

(Publishing, Expansion in MAVEN; Sue, Demonstrate in ACE) and two long-tail types with only 3–5 samples (Change_tool, Breathing in MAVEN; Pardon, Acquit in ACE). We observe that GMMs, by modeling data as a mixture of Gaussian components, capture the multi-modal nature of empirical distributions and concentrate samples in high-density regions, while VAEs project data into a latent space regularized by a Normal distribution $\mathcal{N}(0, I)$ prior, which tends to produce overly smoothed representations.

In the long-tail cases, we observe that GMMs successfully concentrate generated points around the sparse original samples, better approximating their empirical distributions. In contrast, VAE-generated samples tend to collapse toward the center of the embedding space, producing smooth clusters that fail to cover the actual sparse data points. For example, in Change.tool and Pardon, VAE samples cluster in the middle but do not align with the true points, which risks introducing bias. For frequent event types, VAEs tend to produce broadly dispersed samples that cover the general space but smooth out finer substructures, whereas GMMs better preserve the multi-modal clustering patterns present in the original data. This highlights a key distinction: While VAE smoothing may help in dense Gaussian-like categories, it is less effective in capturing fine-grained or non-Gaussian patterns typical in long-tail distributions. Together, these results explain why GMMs provide more faithful approximations of empirical event data, especially under the continual learning setting where long-tail robustness is critical.

Impact of Generated Sample Numbers in SR. To further investigate the role of SR, we evaluate MOZILA under different ratios r , defined as the proportion between the number of generated samples and the size of the replay set. Table 12 presents the results for four values of r . On the MAVEN dataset, the best performance is achieved with $r = 10$, yielding an F1 score of 60.27% on the fifth task, whereas on ACE, the optimal value is $r = 20$, reaching 62.60% on the final task. Although variations in r have limited

Table 12
Ablation results for the number of generated representations by the SR module.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
$r = 20$	62.28	61.78	62.79	61.25	60.02	56.47	56.99	64.44	62.47	62.60
$r = 10$	62.28	61.85	62.92	61.31	60.27	56.47	57.76	64.03	61.34	60.78
$r = 5$	62.28	61.73	62.78	61.03	60.03	56.47	58.95	65.38	62.64	61.16
$r = 2$	62.28	61.86	62.73	60.78	59.38	56.47	58.78	64.68	61.75	60.11

impact in earlier tasks, their influence becomes more pronounced in later stages of training. Importantly, increasing r does not necessarily lead to better performance. This is due to the potential noise introduced in the synthetic samples, stemming from the randomness of the generation process. Such noise may distort the alignment term in MOZILA, potentially misguiding the MOO algorithm and leading to optimization on inaccurate label information. These findings highlight the importance of controlling noise in the generated data to ensure the stability and reliability of synthetic replay.

Impact of Gaussian Component Numbers. Because GMMs are used to generate synthetic samples for previously learned event types in the replay buffer, it is essential to understand how the number of Gaussian components g influences the performance of continual event detection. Table 13 presents the experimental results for different values of g . On the MAVEN dataset, the impact of varying g was relatively minor; although $g = 4$ yielded the highest F1 score, it was only marginally higher—by 0.07%—than the result with $g = 6$ on the fifth task. In contrast, the ACE dataset exhibited greater sensitivity to changes in g . The best result was obtained with $g = 8$, achieving an F1 score of 62.60% after the fifth task. However, increasing g did not consistently lead to better performance. For instance, $g = 4$ resulted in a higher F1 score (62.34%) than $g = 6$, and was only 0.26% lower than the best configuration. These findings suggest that while increasing the number of GMM components may enhance the expressiveness of the generative model, it also introduces a risk of overfitting or noise amplification. Therefore, careful tuning of g is crucial to strike a balance between representation quality and stability in the CED setting.

4.3.4 Error Analysis. To further understand the limitations of our approach, we analyze the errors made by our model and identify two main sources of failure.

Table 13
Ablation results in the number of GMM components.

Task	MAVEN					ACE				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
$g = 2$	62.28	61.69	62.66	60.98	59.78	56.47	57.09	64.42	62.02	60.54
$g = 4$	62.28	61.85	62.92	61.31	60.27	56.47	57.12	64.18	61.63	62.34
$g = 6$	62.28	61.83	62.92	61.28	60.20	56.47	56.31	64.37	62.84	61.54
$g = 8$	62.28	61.82	62.75	61.20	60.20	56.47	56.99	64.44	62.47	62.60

Key Takeaways

- **GMMs are the Most Effective Generative Model:** Among the evaluated generative models, GMMs achieve the best performance and efficiency for SR, outperforming VAEs and cVAEs while avoiding additional computational overhead.
- **Hyperparameter Sensitivity Requires Careful Tuning:** The number of Gaussian components and generated samples significantly affects SR performance, especially in later tasks, highlighting the need for careful tuning to balance representation quality and noise control.

Misclassification into the NA Class. A recurring challenge in event detection is the overwhelming prevalence of the NA class, which represents spans without any annotated event. This class imbalance causes the model to bias toward NA predictions, even when triggers strongly indicate a valid event type. After training on the last task and evaluating on the full ontology, we find that 18.25% of true event triggers are misclassified as NA on MAVEN and 33.54% on ACE. Although this error rate remains non-trivial, it is markedly lower than KT, which misclassifies 51.23% on MAVEN and 45.95% on ACE. Similar challenge and pattern are also pronounced in RAMS and GENEVA. This improvement demonstrates that our method better preserves event-specific knowledge and reduces the model’s tendency to overfit to NA. However, the residual NA misclassifications suggest that future research may need to incorporate strategies such as adaptive rebalancing or more explicit modeling of the NA distribution to further mitigate this issue.

Confusion Among Semantically Similar Event Types. The confusion matrices in Figure 9 highlight that errors in event detection often occur between semantically related categories, where distinctions are subtle or highly context-dependent. Examples of failure cases are provided in Table 14. In the MAVEN dataset, *Military_operation* and *Hostile_encounter* are frequently confused, as illustrated by the Example 1 where the trigger “war” was misclassified. This confusion stems from the overlapping semantics of organized warfare versus localized encounters, particularly when the trigger term is generic. Another notable confusion occurs between *Arriving* and *Achieve*, as seen in the Example 3. Here, the trigger reached could imply a spatial arrival or the attainment of a state, showing how polysemous verbs blur category boundaries. Similarly, *Self_motion* versus *Motion* errors are evident in sentences like “moved towards Riga”, where the model misinterprets the distinction between an agent’s autonomous movement and broader motion events.

In the ACE dataset, we observe analogous issues. The categories *Attack* and *Die* are often entangled, as demonstrated by the misclassification in Example 4. The trigger “killings” simultaneously encodes the act of violent aggression (Attack) and the outcome

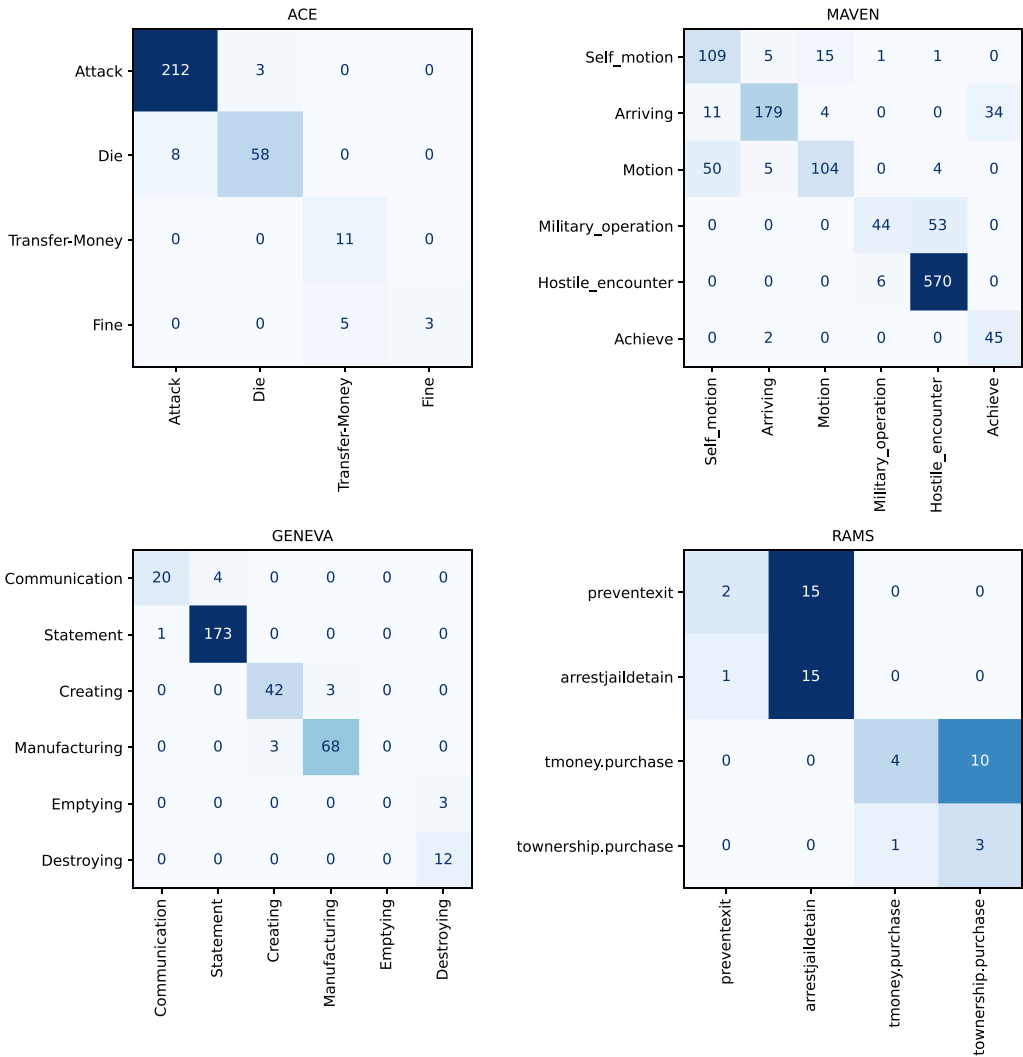


Figure 9 Confusion matrices for event detection across the four evaluated datasets. The figure visualizes the most confusing category pairs for MOZILA.

of death (Die), leading to inherent ambiguity. Another confusion appears between *Fine* and *Transfer-Money*, such as in “Taco Bell has to pay the true creators...”. While pay denotes a monetary transaction, in legal contexts it often implies a fine, requiring deeper contextual modeling. Similar patterns are observed in GENEVA, where semantically adjacent event types; such as *Manufacturing* vs. *Creating*, *Communication* vs. *Statement*, and *Destroying* vs. *Emptying*, exhibit substantial overlap in surface triggers but differ in subtle semantic intent.

In RAMS, semantic confusion is further amplified by the dataset’s extremely fine-grained and hierarchical event ontology. Many event types share nearly identical surface triggers and differ only in subtle distinctions at the parent or end-state level. For

Table 14
Examples of confusion between semantically similar event types in CED. The table lists misclassified triggers along with their predicted and gold labels. The highlighted word indicates the event trigger.

ID	Sentence & Trigger	Prediction	Label
1	After the capture of Pärnu, Jan Karol Chodkiewicz, commander of the Lithuanian army, moved towards Riga, which was occupied by the Swedes under the command of Joachim Frederick von Mansfeld.	Motion	Self_Motion
2	The war between the two leagues restarted in 431 BC, leading to the Second Peloponnesian War.	Hostile_encounter	Military_operation
3	The temperature reached 104 degrees that day (over 120 degrees on the field).	Achieve	Arriving
4	It's true, Hentoff said, that the U. N. issued a report in 1999 that acknowledged, to some extent, its role in the killings .	Die	Attack
5	The jury has handed down a \$30 million, a Grand Rapids jury saying Taco Bell has to pay the true creators of the Chihuahua mascot.	Transfer-Money	Fine
6	No evidence obtained to date suggests that Al Qa'ida has actually succeeded in producing significant quantities of chemical or biological agents or developed an effective means of delivery, but the intention to do so is clearly present.	Manufacturing	Creating
7	I think we know what we're going to speak about.	Communication	Statement
8	The South African experience demonstrates that at least under some conditions, unilateral disarmament is not only possible, but can improve a nation's security.	Destroying	Emptying
9	The other remains encrypted and is the subject of an online auction , payable in bitcoin, the cryptocurrency.	transaction .transferownership .purchase	transaction .transfermoney .purchase
10	By 2002, that number had jumped to 24,162, but by 2012, the number imprisoned had dropped to 22,558.	justice .arrestjaildetain .arrestjaildetain	movement .transportperson .preventexit

example, in the sentence “The other remains encrypted and is the subject of an online auction, payable in bitcoin, the cryptocurrency,” the trigger *auction* simultaneously implies a monetary transaction and a change of ownership. While the final event is annotated as *Purchase* under *Transfer-Ownership*, the strong lexical association with payment (e.g., *payable in bitcoin*) biases the model toward *Transfer-Money*. Disambiguating such cases requires reasoning about the intended outcome of the event rather than surface-level financial cues. Similarly, end-state events such as *Arrest-Jail-Detain* and *Prevent-Exit* often co-occur in closely related contexts, yet differ in the intended outcome (detention versus restriction of movement). These distinctions require fine-grained semantic and contextual reasoning beyond lexical cues, making RAMS particularly susceptible to semantic confusion.

Importantly, these confusing pairs typically lie in different training stages of CED. This sequential learning setup may cause the model to lose discriminative information acquired earlier, limiting its ability to effectively separate closely related categories. While this phenomenon has not been extensively examined in CED, it has been critically discussed in related areas like continual relation extraction, where sequential training exacerbates semantic overlap and forgetting challenges (Wang et al. 2022; Zhao et al. 2023; Le et al. 2025). Together, these cases reveal that many errors stem not from random

noise but from systematic ambiguities in event detection combined with continual learning challenges. Addressing this requires not only richer contextual modeling but also strategies explicitly designed to preserve fine-grained distinctions across stages.

Impact of LMHP on Semantic Confusion. While LMHP is designed to preserve general linguistic knowledge from the pretrained language model, an overly strong constraint on the shared LM head may potentially bias the model toward coarse-grained semantic representations, which could increase confusion among semantically related event types. To further examine the potential negative effect of LMHP on semantic confusion, we define **Semantic Error Rate (SER)** as the probability that a test instance belonging to a target event class is incorrectly predicted as a semantically related event type. As a result, SER allows us to directly answer the question: *Given an instance from a target event class, how likely is the model to confuse it with a semantically similar event type?*

Formally, given a confusion matrix $C \in \mathbb{R}^{K \times K}$, let $\mathcal{C}$ denote the set of event classes under consideration (excluding NA), and let $\mathcal{S}(i)$ denote the set of semantically similar event types to class i . SER is defined as

$$\text{SER} = \frac{\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{S}(i)} C_{i,j}}{\sum_{i \in \mathcal{C}} |i|} \quad (34)$$

where $|i|$ denotes the number of ground-truth instances of class i . To focus the analysis on the most prominent semantic confusions, we compute SER on a restricted set of event classes derived from the confusion matrix. Specifically, we first identify the top-10 most frequent off-diagonal confusion pairs and define the considered class set $\mathcal{C}$ as the union of event types involved in these pairs. SER is then computed only over instances whose ground-truth labels belong to $\mathcal{C}$, ensuring that the metric emphasizes dominant and semantically meaningful confusions.

Figure 10 presents both quantitative and qualitative evidence on the impact of LMHP on semantic confusion. As shown in the left panel, LMHP leads to a small increase in SER on MAVEN, GENEVA, and RAMS, while slightly reducing SER on ACE. Importantly, the magnitude of these changes is small, indicating that LMHP does

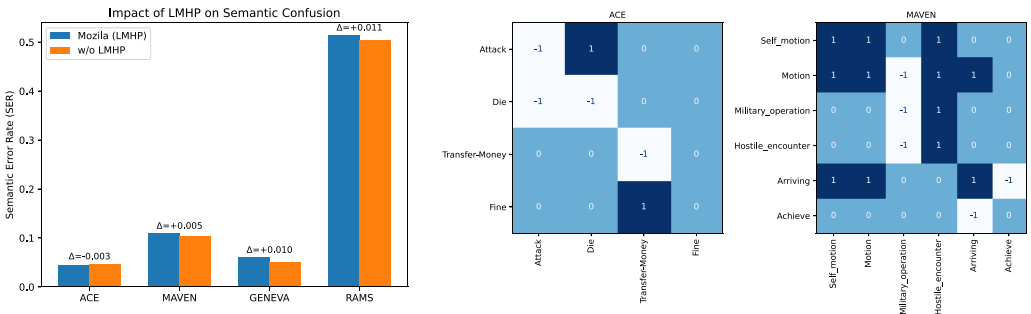


Figure 10 Impact of LMHP on semantic confusion. **Left:** SER comparison between the full model (MOZILA, with LMHP) and its ablation without LMHP across four datasets; Δ denotes the SER difference. **Right:** Signed confusion matrix differences (MOZILA – w/o LMHP) for representative semantically similar event types on ACE and MAVEN, where positive (negative) values indicate increased (reduced) confusion under LMHP.

not substantially amplify semantic confusion, but instead introduces a limited trade-off between preserving linguistic knowledge and fine-grained semantic discrimination. This effect is more pronounced on datasets with high semantic overlap (e.g., RAMS with $|\Delta| = 0.011$), where event boundaries are inherently ambiguous and even minor shifts in representation can influence semantic confusability. The signed confusion matrix differences in the right panel further reveal that SER variations stem from localized changes among a small set of semantically adjacent event pairs rather than a global degradation across classes. Specifically, LMHP slightly increases confusion among closely related *motion*- events in MAVEN, while reducing or mixing confusion for certain fine-grained pairs (e.g., *Attack* vs. *Die* in ACE and *Arriving* vs. *Achieve* in MAVEN). This asymmetric pattern suggests that LMHP does not uniformly blur semantic distinctions; instead, it selectively affects event types that already exhibit strong semantic overlap.

Overall, these observations indicate that LMHP introduces a controlled and interpretable semantic trade-off. While it may marginally increase confusion among inherently ambiguous event types, it does not dominate the model’s error profile and remains compatible with improved robustness and overall performance.

Key Takeaways

- Misclassification into the NA class remains a critical challenge, largely driven by class imbalance.
- Confusion frequently occurs between semantically similar event types, often introduced in different training stages, revealing systematic ambiguities in event detection and the difficulty of preserving fine-grained distinctions under continual learning.
- Constraining linguistic knowledge through LMHP may slightly increase semantic confusion when fine-tuning on domain-specific datasets; however, this trade-off remains limited in magnitude and is outweighed by consistent gains in performance and robustness across datasets.

5. Conclusion and Future Works

In this article, we propose MOZILA, a novel and robust framework for Continual Event Detection (CED) that integrates two key components: Language Modeling Head Preservation (LMHP) and Sharpness-Aware Multi-Objective Optimization (SaMOO). To address the dual challenges of representation degradation and objective conflict in continual learning, our approach leverages alignment techniques to retain pre-trained linguistic knowledge and adapt gradient-based MOO strategies with Sharpness-Aware Minimization (SAM) to enhance generalization and stability across tasks. Additionally, we introduce synthetic representations to mitigate data imbalance in memory-based

replay, serving as a crucial component that supports the effectiveness of the proposed LMHP and SaMOO modules. We conduct extensive experiments on four benchmarks: ACE, MAVEN, and the newly introduced RAMS and GENEVA datasets, covering diverse event ontologies, sentence lengths, and degrees of semantic overlap. Across all datasets, MOZILA consistently outperforms strong baselines, demonstrating robust performance under increasingly challenging continual learning scenarios. Ablation studies further validate the effectiveness of our design, including insights into hyperparameter sensitivity and the choice of generative models for synthetic replay.

For future work, we will address several limitations of MOZILA. While Synthetic Representation (SR) effectively mitigates class imbalance, it may introduce noise, motivating future exploration of more robust generation or denoising strategies. The current SaMOO design also involves repeated gradient computations, leading to high training costs; thus, more efficient optimization techniques are worth investigating. Although our evaluation focuses on Event Detection, MOZILA is task-agnostic and applicable to other NLP tasks such as Named Entity Recognition and Relation Extraction, which we plan to explore. Additionally, we aim to extend our framework to examine its integration with large language models to handle more complex event ontologies.

Acknowledgments

This project is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2025.16. We would like to express our sincere gratitude to Thanh-Thien Le, Viet Dao, Linh Van Nguyen, Thi-Nhung Nguyen, Van-Cuong Pham, Quyen Tran, and Thien Huu Nguyen for their valuable contributions to the prior work that laid the foundation for this research. Although they are not directly involved in the current study, their earlier efforts and insights have significantly shaped the direction of this work. We deeply appreciate their indirect support and the groundwork they have established in the field of continual event detection.

References

- Ahn, Hongjoon, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. *Advances in Neural Information Processing Systems*, 32.
- Aljundi, Rahaf, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154. https://doi.org/10.1007/978-3-030-01219-9_9
- Aljundi, Rahaf, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert Gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375. <https://doi.org/10.1109/CVPR.2017.753>
- Ban, Hao and Kaiyi Ji. 2024. Fair resource allocation in multi-task learning. *arXiv preprint arXiv:2402.15638*.
- Bhat, Prashant, Bahram Zonooz, and Elahe Arani. 2023. Task-aware information routing from common representation space in lifelong learning. *arXiv preprint arXiv:2302.11346*.
- Cao, Pengfei, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 707–717. <https://doi.org/10.18653/v1/2020.emnlp-main.52>
- Caramia, Massimiliano, Paolo Dell’Olmo, Massimiliano Caramia, and Paolo Dell’Olmo. 2020. Multi-objective optimization. In *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level, Sustainability, and Safety with Optimization Algorithms*, pages 21–51. Springer. https://doi.org/10.1007/978-3-030-50812-8_4
- Castro, Francisco M., Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248. https://doi.org/10.1007/978-3-030-01258-8_15

- Chaudhry, Arslan, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*.
- Chen, Ruirui, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17772–17780. <https://doi.org/10.1609/aaai.v38i16.29730>
- Chen, Zhiyuan and Bing Liu. 2018. *Lifelong Machine Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer. <https://doi.org/10.1007/978-3-031-01581-6>
- Cliff, Norman. 1993. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494. <https://doi.org/10.1037/0033-2909.114.3.494>
- Cuturi, Marco. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300.
- Dao, Bao-Ngoc, Minh Le, Quang Nguyen, Luyen Ngo Dinh, Nam Le Hai, and Linh Ngo Van. 2026. WAVE++: Capturing within-task variance for continual relation extraction with adaptive prompting. *Neurocomputing*, 675:132915. <https://doi.org/10.1016/j.neucom.2026.132915>
- Dao, Viet, Van-Cuong Pham, Quyen Tran, Thanh-Thien Le, Linh Van Ngo, and Thien Huu Nguyen. 2024. Lifelong event detection via optimal transport. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12610–12621. <https://doi.org/10.18653/v1/2024.emnlp-main.701>
- de Masson D’Autume, Cyprien, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.
- Devlin, Jacob, 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.
- Ebner, Seth, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077. <https://doi.org/10.18653/v1/2020.ac1-main.718>
- Foret, Pierre, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*.
- French, Robert M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135. [https://doi.org/10.1016/S1364-6613\(99\)01294-2](https://doi.org/10.1016/S1364-6613(99)01294-2), PubMed: 10322466
- Frogner, Charlie, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A. Poggio. 2015. Learning with a Wasserstein loss. *Advances in Neural Information Processing Systems*, 28.
- Ghose, Abhishek and Emma Nguyen. 2024. On the fragility of active learners for text classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22217–22233. <https://doi.org/10.18653/v1/2024.emnlp-main.1240>
- Gunantara, Nyoman. 2018. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5. <https://doi.org/10.1080/23311916.2018.1502242>
- Guo, Yiduo, Bing Liu, and Dongyan Zhao. 2022. Online continual learning through mutual information maximization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8109–8126.
- Hai, Nam Le, Trang Nguyen, Linh Ngo Van, Thien Huu Nguyen, and Khoat Than. 2024. Continual variational dropout: A view of auxiliary local variables in continual learning. *Machine Learning*, 113:281–323. <https://doi.org/10.1007/s10994-023-06487-7>
- He, Yifei, Shiji Zhou, Guojun Zhang, Hyokun Yun, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Han Zhao. 2024. Robust multi-task learning with excess risks. *arXiv preprint arXiv:2402.02009*.
- Johnson, Justin M. and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6:1–54. <https://doi.org/10.1186/s40537-019-0192-5>

- Kingma, Diederik P. and Max Welling. 2014. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations*.
- Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Le, Anh Duc, Nam Le Hai, Thanh Xuan Nguyen, Linh Ngo Van, Nguyen Thi Ngoc Diep, Sang Dinh, and Thien Huu Nguyen. 2025. Enhancing discriminative representation in similar relation clusters for few-shot continual relation extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2450–2467. <https://doi.org/10.18653/v1/2025.naacl-long.123>
- Le, Thanh-Thien, Viet Dao, Linh Nguyen, Thi-Nhung Nguyen, Linh Ngo, and Thien Nguyen. 2024. SharpSeq: Empowering continual event detection through sharpness-aware sequential-task learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3632–3644. <https://doi.org/10.18653/v1/2024.naacl-long.200>
- Li, Qian, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.
- Li, Xilai, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934.
- Liu, Bo, Yihao Feng, Peter Stone, and Qiang Liu. 2024. FAMO: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-2500>
- Liu, Bo, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021a. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890.
- Liu, Liyang, Yi Li, Zhanghui Kuang, J. Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021b. Towards impartial multi-task learning. In *9th International Conference on Learning Representations*.
- Liu, Minqian, Shiyu Chang, and Lifu Huang. 2022. Incremental prompting: Episodic memory prompt for lifelong event detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2157–2165.
- Lopez-Paz, David and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30.
- Loshchilov, Ilya and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*.
- Ma, Shengkun, Jiale Han, Yi Liang, and Bo Cheng. 2024. Making pre-trained language models better continual few-shot relation extractors. *arXiv preprint arXiv:2402.15713*. <https://doi.org/10.63317/3ezkbexw5upj>
- Man, Hieu, Nghia Trung Ngo, Linh Ngo Van, and Thien Huu Nguyen. 2022. Selecting optimal context sentences for event-event relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11058–11066. <https://doi.org/10.1609/aaai.v36i10.21354>
- Mirzadeh, Seyed Iman, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. 2020. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320.
- Nash, John F. 1953. Two-person cooperative games. *Econometrica*, 21:128–140. <https://doi.org/10.2307/1906951>
- Navon, Aviv, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*.
- Nguyen, Chien, Linh Ngo, and Thien Nguyen. 2023. Retrieving relevant context to align representations for cross-lingual event detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2157–2170.
- Nguyen, Cuong V., Yingzhen Li, Thang D. Bui, and Richard E. Turner. 2017. Variational continual learning. *arXiv preprint arXiv:1710.10628*.

- Parekh, Tanmay, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023. GENEVA: Benchmarking generalizability for event argument extraction with hundreds of event types and argument roles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3664–3686.
- Parekh, Tanmay, Anh Mac, Jiarui Yu, Yuxuan Dong, Syed Shahriar, Bonnie Liu, Eric Yang, Kuan-Hao Huang, Wei Wang, Nanyun Peng, et al. 2024. Event detection from social media for epidemic prediction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5758–5783. <https://doi.org/10.18653/v1/2024.naacl-long.322>
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- Pham, Thanh Duc, Nam Le Hai, Linh Ngo Van, Nguyen Thi Ngoc Diep, Sang Dinh, and Thien Huu Nguyen. 2025. Mitigating non-representative prototypes and representation bias in few-shot continual relation extraction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10791–10809. <https://doi.org/10.18653/v1/2025.acl-long.530>
- Phan, Hoang, Lam Tran, Ngoc N. Tran, Nhat Ho, Dinh Phung, and Trung Le. 2023. Improving multi-task learning via seeking task-based flat regions. *arXiv preprint arxiv:2211.13723*.
- Qin, Chengwei, Ruirui Chen, Ruochen Zhao, Wenhan Xia, and Shafiq Joty. 2024. Lifelong event detection with embedding space separation and compaction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 594–602. <https://doi.org/10.18653/v1/2024.naacl-short.50>
- Rebuffi, Sylvestre Alvisé, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5533–5542. <https://doi.org/10.1109/CVPR.2017.587>
- Reynolds, Douglas A., et al. 2009. Gaussian mixture models. In S. Z. Li, and A. Jain, editors, *Encyclopedia of Biometrics*, pages 659–663. Springer. https://doi.org/10.1007/978-0-387-73003-5_196
- Sener, Ozan and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31.
- Serra, Joan, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557.
- Sohn, Kihyuk, Honglak Lee, and Xinchun Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28.
- Song, Zhiyi, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From Light to Rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98.
- Sun, Zhicheng, Yadong Mu, and Gang Hua. 2023. Regularizing second-order influences for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20166–20175. <https://doi.org/10.1109/CVPR52729.2023.01931>
- Thrun, Sebastian and Tom M. Mitchell. 1995. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1–2):25–46. [https://doi.org/10.1016/0921-8890\(95\)00004-Y](https://doi.org/10.1016/0921-8890(95)00004-Y)
- Tran, Quyen, Nguyen Xuan Thanh, Nguyen Hoang Anh, Nam Le Hai, Trung Le, Linh Van Ngo, and Thien Huu Nguyen. 2024. Preserving generalization of language models in few-shot continual relation extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13771–13784. <https://doi.org/10.18653/v1/2024.emnlp-main.763>
- van den Oord, Aäron, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.
- Veyseh, Amir Pouran Ben, Javid Ebrahimi, Franck Dernoncourt, and Thien Nguyen.

2022. MEE: A novel multilingual event extraction dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9603–9613. <https://doi.org/10.18653/v1/2022.emnlp-main.652>
- Wadden, David, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5783–5788. <https://doi.org/10.18653/v1/D19-1585>
- Walker, Christopher, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus LDC2006T06. Web Download. Linguistic Data Consortium.
- Wang, Liyuan, Xingxing Zhang, Qian Li, Mingtian Zhang, Hang Su, Jun Zhu, and Yi Zhong. 2023. Incorporating neuro-inspired adaptability for continual learning in artificial intelligence. *Nature Machine Intelligence*, 5(12):1356–1368. <https://doi.org/10.1038/s42256-023-00747-w>
- Wang, Peiyi, Yifan Song, Tianyu Liu, Binghuai Lin, Yunbo Cao, Sujian Li, and Zhifang Sui. 2022. Learning robust representations for continual relation extraction via adversarial class augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6264–6278. <https://doi.org/10.18653/v1/2022.emnlp-main.420>
- Wang, Xiaozhi, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. Maven: A massive general domain event detection dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1652–1671. <https://doi.org/10.18653/v1/2020.emnlp-main.129>
- Wang, Zitao, Xinyi Wang, and Wei Hu. 2023. Continual event extraction with semantic confusion rectification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11945–11955. <https://doi.org/10.18653/v1/2023.emnlp-main.732>
- Welling, Max. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128. <https://doi.org/10.1145/1553374.1553517>
- Wilcoxon, Frank. 1992. Individual comparisons by ranking methods. *Breakthroughs in Statistics: Methodology and Distribution*, pages 196–202. Springer. https://doi.org/10.1007/978-1-4612-4380-9_16
- Wu, Yue, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382. <https://doi.org/10.1109/CVPR.2019.00046>
- Yu, Pengfei, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5278–5290. <https://doi.org/10.18653/v1/2021.emnlp-main.428>
- Yu, Tianhe, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.
- Zhao, He, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. 2021. Neural topic model via optimal transport. In *International Conference on Learning Representations*.
- Zhao, Wenzheng, Yuanming Cui, and Wei Hu. 2023. Improving continual relation extraction by distinguishing analogous semantics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1162–1175.
- Zhao, Xuyang, Huiyuan Wang, Weiran Huang, and Wei Lin. 2024. A statistical theory of regularization-based continual learning. *arXiv preprint arXiv:2406.06213*.
- Zhou, Da Wei, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. 2024. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564. <https://doi.org/10.1109/CVPR52733.2024.02223>