Strategic Optimization for Worst-Case Augmentation and Classification

Jen-Tzung Chien, *Senior Member, IEEE*, Mahdin Rohmatillah, and Chang-Ting Chu

***Abstract*—Adversarial data augmentation techniques have recently demonstrated potential in enhancing the robustness of machine learning models by identifying potential worst-case data augmentation. However, most of the existing methods implemented a single augmentation strategy for different instances through a process of greedy search, resulting in suboptimal quality of the generated data. Furthermore, previous studies that incorporated reinforcement learning (RL) to apply unique augmentation strategies required high computational cost, as it necessitated a child network to compute the reward during the optimization process. Given these limitations, this study introduces a strategic adversarial data augmentation approach that leverages RL to search for and emulate the worst-case variations through a sequence of augmentation actions. By defining a reward function with an information-theoretic perspective along with the proper definition of state space, a proficient strategy for stacking multiple augmentation strategies can be carried out in an inexpensive way and can be smoothly integrated into classifier training, thereby enhancing model robustness against unseen noises. The proposed adversarial training method was evaluated on ten different types of unseen human-readable noises across six distinct text classification tasks. Experimental results indicate that the proposed method significantly improves model robustness in compensating for unseen noises.**

***Index Terms*—Adversarial learning, model robustness, policy optimization, text classification, worst-case augmentation.**

I. Introduction

FINE-TUNING approach has been considered as one of the most important techniques in deep learning areas. Most of the state-of-the-art result (SOTA) results were achieved by utilizing this method which is initiated by building a strong model which has been trained from a very large dataset, then the learned model is fine-tuned to any downstream task. In the natural language processing (NLP) domain, the pre-trained language models (PLMs) are mostly built by using transformers [1], [2], [3], [4], trained with an abundance of corpus taken from various datasets. However, the robustness of PLMs in the NLP domain presents a significant challenge. Generating attacks for NLP models is relatively straightforward compared to computer vision (CV) models, as minor modifications to the original sentence, such as word elimination or changing singular to plural words, can alter the classifier prediction. This issue is further compounded by the assumption in traditional learning procedures that training and test data distributions are identical. In contrast, noisy input data from different distributions are common in the model implementation. Several studies have demonstrated that PLMs struggle to handle unintended noisy inputs and specifically designed attacks based on certain augmentation methods, such as synonym substitution [5], [6], or character-level augmentation [7], [8], resulting in significant degradation in model accuracy. Interestingly, humans can accurately interpret the meanings of sentences even with the same generated perturbations. This discrepancy reflects the need for improving the robustness in PLMs, particularly in handling the unseen noises and adversarial attacks [9].

To enhance the robustness of the model, it is crucial to meticulously design adversarial training, which incorporates both *attack* and *defense* strategies. This can be achieved through the generation and utilization of augmented instances, where exploration and exploitation are respectively considered. The *attack* strategy through adversarial data augmentation (ADA) have been proposed to bolster model robustness in the fields such as machine translation [10] and text classification [6], [7], [8], [11], [12], [13]. However, these methods employed a greedy search and required heuristic rules during the search process to generate adversarial instances, which could potentially constrain the quality of using adversarial data for robustness optimization. Furthermore, the implementation of greedy search will limit the variation of adversarial instances generated from an identical input sentence. Other works have proposed a series of augmentation actions utilizing reinforcement learning (RL) approaches [14], [15], inspired by the AutoAugment [16] paradigm that adheres to the neural architecture search (NAS) framework. However, the necessity for either a child network to generate reward signals or external knowledge to define the action space in an RL environment [17] could potentially cause the computational issues and hinder the performance of the augmentation policy. In an attempt to alleviate this complexity, a fixed stack of augmentation actions has been proposed [18]. Despite demonstrating the promising performance, fixing the strategies to the whole data could not really simulate the worst-case data augmentation scenario.

This study aims to develop an efficient and effective learning algorithm that enhances the model robustness. This algorithm

Received 16 April 2023; revised 14 January 2024 and 27 July 2024; accepted 26 November 2024. Date of publication 4 December 2024; date of current version 24 January 2025. This work was supported by the National Science and Technology Council, Taiwan, under Grant NSTC 113-2634-F-A49-006. The associate editor coordinating the review of this article and approving it for publication was Dr. Jing Huang. *(Corresponding author: Jen-Tzung Chien.)*

Jen-Tzung Chien and Chang-Ting Chu are with the Institute of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: jtchien@nycu.edu.tw).

Mahdin Rohmatillah is with the Department of Electrical Engineering, Universitas Brawijaya, Kota Malang 65145, Indonesia.

Digital Object Identifier 10.1109/TASLP.2024.3511264

involves the optimal search for distinctive stacking augmentation strategies, referred to as the *attack* strategies, in individual data points. These are paired with the corresponding *defense* strategies [19] to optimally utilize the augmented data, thereby improving model performance against unseen noises. In particular, a novel RL algorithm, based on REINFORCE [20], is reformulated to address the exponential-growth search space resulting from the distinctive augmentation strategies. Inspired from an enhanced variant of ADA, which has demonstrated success in the field of computer vision [21], this paper introduces an information-theoretic reward setting derived from information bottleneck (IB) principle for finding the optimal *attack* strategies. The issue of generalization by compressing the neural model [22] is addressed. Therefore, the adversarial learning scenario incorporates not only the cross-entropy loss for text classification but also the uncertainty modeling from IB principle to address the generalization to unseen noises. For the *defense* strategy, in addition to IB implementation, two regularization objectives based on the Jensen-Shannon divergence and the supervised contrastive loss are proposed to meet local and global constraints, respectively. The efficacy of the proposed method is demonstrated by evaluating different types of unseen perturbations to test the robustness of target model in various text classification tasks.

The contributions of this work can be summarized as follows: Firstly, a novel adversarial data augmentation is proposed, which involves distinctive transformation strategies for individual data, thereby enhancing the diversity of augmented data. Secondly, a new strategic optimization is introduced to address the exponential-growth search space resulting from the potential for various transformations, while maintaining the computational feasibility. Thirdly, a new information-theoretic reward setting for the RL policy to find the best augmentation strategies is derived, along with additional constraints to provide meaningful feedback during the policy optimization. Lastly, a range of evaluation sets involving different noises are examined to demonstrate the robustness of the model which is achieved through the strategic *attack* and *defense*.

The presentation of this work is organized as follows. Section II addresses the fundamentals of worst-case generalization. Section III surveys the solutions to strategic and adversarial augmentation. Section IV details the worst-case augmentation where the adversarial and informative *attack* and *defense* are performed. Section V reports a series of experiments with the elaborated robustness evaluation and behavior analysis. The findings of this work are summarized in Section VI.

II. Worst-Case Generalization

Fig. 1 illustrates the concept of worst-case generalization. Given a set of clean training data from source distribution P_0, the goal is to generate the examples that tackle the worst-case augmentation problem to expand the model coverage for possible unseen noisy inputs. For example, T_1 might be composed of spelling errors of some clean data distributed by P_0, and T_2 might be another kind of noisy input such as back-translation error. In order to train a model in a single source domain P_0 [23] and extend it to an unseen noisy data domain P [24], this study formulates the following worst-case problem in a form of minimax optimization

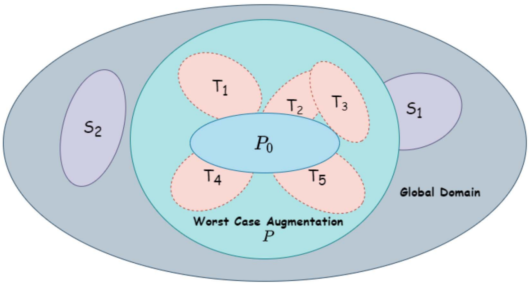


Fig. 1. Illustration for single domain generalization for unseen noisy inputs following the worst-case augmentation P. P_0 is the clean data distribution. Each T_i is a different type of noisy input that happens in a target domain. S_i represents different unseen source data in a global domain.

$$\underset{\theta\in\Theta}{\text{minimize}}\left\{\sup_{P}\left\{\mathbb{E}[\mathcal{L}(X,Y;\theta)]\text{ s.t. }D_\theta\left(P_0,P\right)\leq d\right\}\right\}\quad(1)$$

where $\theta\in\Theta$ is the parameter of a target classifier, $X\in\mathcal{X}$ and $Y\in\mathcal{Y}$ are the data points and the corresponding labels of a training set $\mathcal{D}=\{X,Y\}$, respectively, $\mathcal{L}:(\mathcal{X},\mathcal{Y})\times\Theta\rightarrow\mathbb{R}$ is the loss function for worst-case problem, and D_θ represents the distance metric. The solution to this problem would like to assure that the resulting model maintains its performance on the original data distribution P_0 when dealing with the noisy data distribution P which is a neighbor away from P_0 within a distance bound d. The distance metric is defined in a latent semantic space. Even though the text inputs are far away from their original appearances, a model gained by the worst-case data augmentation will be robust and representative.

In general, Wasserstein metric can be used as the distance metric D_θ to measure the distance in latent semantic space $\mathcal{Z}$ [25]. Using neural network as a target classifier, the parameters $\theta=\{\theta_e,\theta_c\}$ consist of θ_e for encoding a raw text into an embedding as well as θ_c for finding the outputs from classification layer. Let $c_\theta:\mathcal{Z}\times\mathcal{Z}\rightarrow\mathbb{R}_+\cup\{\infty\}$ denote the positive transportation cost for moving the mass from original data point $(\mathbf{x}_0,\mathbf{y}_0)$ to the augmented data point $(\mathbf{x},\mathbf{y})$ which is measured by their latent embeddings $\mathbf{z}_0=f(\mathbf{x}_0;\theta_e)$ and $\mathbf{z}=f(\mathbf{x};\theta_e)$. This cost is infinite if the label $\mathbf{y}$ differs from its original label $\mathbf{y}_0$, and is defined as

$$c_\theta((\mathbf{x}_0,\mathbf{y}_0),(\mathbf{x},\mathbf{y}))=\frac{1}{2}\|\mathbf{z}_0-\mathbf{z}\|_2^2+\infty\cdot\mathbb{1}\{\mathbf{y}_0\neq\mathbf{y}\}\quad(2)$$

where $\mathbb{1}\{\cdot\}$ is an indicator. Considering the probability masses P and P_0 represented in semantic space $\mathcal{Z}$, the distance metric D_θ can be expressed as

$$D_\theta(P_0,P)=\inf_{M\in\mathcal{M}(P_0,P)}\mathbb{E}_M[c_\theta((X_0,Y_0),(X,Y))]\quad(3)$$

where the expectation is operated over the greatest lower bound of probability couplings $M\in\mathcal{M}(P_0,P)$. Unfortunately, the supremum over probability distributions in (1) is computationally intractable. Then, the Lagrangian relaxation with a penalty parameter ρ is applied to rewrite it as

$$\underset{\theta\in\Theta}{\text{minimize}}\left\{\sup_{P}\left\{\mathbb{E}_P[\mathcal{L}(X,Y;\theta)]-\rho D_\theta(P_0,P)\right\}\right\}. \quad (4)$$

This paper presents a strategic optimization to build an augmented target distribution P around P_0 according to (4) by searching for a series of augmentation strategies which is different from the previous works [5], [6], [7], [8] where the greedy method was used to identify a single augmentation strategy.

III. Auto Augmentation

This paper presents strategic and adversarial learning for data augmentation which tackles the optimization problem for worst-case generalization. Some related works are surveyed.

A. Strategic Augmentation

Designing a proper strategy for data augmentation in natural language tasks has emerged as a compelling research area, driven by the aspiration to replicate the success of strategic augmentation demonstrated in computer vision (CV) tasks [16], [26]. Over recent years, numerous studies have endeavored to leverage established techniques in text augmentation, such as back-translation [27], data noising [28], and easy data augmentation (EDA) [29], as their primary augmentation strategies. However, the findings, as evidenced in [30], have only shown suboptimal performance due to the simplistic nature of their learning processes. These studies predominantly employed the self-supervised augmentation, utilizing either a distance metric or a singular augmentation policy like back-translation. In addition, the other research has proposed a series of augmentation actions using reinforcement learning (RL) methods [14], [15], drawing inspiration from the AutoAugment [16] paradigm, which conforms to the neural architecture search (NAS) framework. Nevertheless, the requirement for either a child network to generate reward signals or an external knowledge to define the action space in an RL environment could potentially result in computational challenges and limit the performance of the augmentation policy. To mitigate the issue of computation complexity, a fixed stack of augmentation actions has been suggested [18]. Although this approach has shown promising performance, applying the fixed strategies to the entire dataset may not be effective in mimicking the worst-case data augmentation scenario.

B. Adversarial Augmentation

Adversarial data augmentation (ADA) has been investigated thoroughly in recent years as a means to improve model robustness for specific adversarial attack. ADA aims to expand the search space or coverage area by introducing additional meaningful data during model training to overcome adversarial attack that possibly occurs in real-world applications. Previously, most studies have employed a greedy search scheme to identify the adversarial data [6], [7], [8], [13], [31]. Although this scheme can generate potential adversarial instances, the deployment of the greedy search restricts the exploration for worst-case data augmentation. This is due to the deterministic nature of the greedy method, which invariably produces identical outputs given identical inputs. However, in a worst-case scenario, multiple adversarial examples may originate from the same input. Several previous greedy-based methods also necessitated the pre-defined heuristic and linguistic rules [11], [12], potentially escalating the time complexity of these greedy methods. Alternative strategies have utilized the adversarial training to develop a rewriter model [32], designed to manipulate inputs to facilitate accurate classification by the classifier. Regrettably, the incorporation of an additional text rewriter model can increase inference time, particularly when the text rewriter model is constructed with large language models (LLMs) such as T5 [33]. Additionally, some of the previous works have also attempted to enhance model robustness through representational augmentation by giving perturbation in the embedding level [34], [35]. Unfortunately, it may lose the original semantic interpretability. A further limitation of previous studies is that the majority of works were evaluated by using the same attack procedure that was introduced during adversarial training [35]. In contrast, in practical applications, a variety of noise types may attack the model, suggesting that model evaluations should consider a range of unseen noises. In contrast to these conventional approaches, this study proposes a novel method that contemplates a unique sequence of augmentation strategies for each data point, thereby enhancing the quality and diversity of the generated adversarial data. To decrease computational complexity while enlarging the search space of potential adversarial augmentation strategies for each instance, a learning algorithm predicated on RL is introduced. By defining a reward function in accordance with the IB principle, the worst-case augmentation is implemented without requiring any child network. Furthermore, it is crucial to examine an algorithm capable of not only addressing specific attacks but also enhancing the robustness of a model against unseen noises. Therefore, a variety of diverse evaluations considering different kinds of unseen readable noises are proposed to emulate real-world scenarios.

IV. Worst-Case Augmentation

This study presents an implementation of a worst-case augmentation strategy through a systematic exploration of various augmentation strategies. The proposed augmenter, which employs the REINFORCE algorithm [20], is depicted at the top of Fig. 2. The policy or the REINFORCE augmenter with parameter θ_a arranges a series of augmentation operations to generate the worst-case examples. The quality of the generated adversarial data is optimized by maximizing the cumulative reward based on the IB principle. Following the completion of the REINFORCE augmenter training, as illustrated at the bottom of Fig. 2, the generated data are employed to enrich the generalization and robustness of the target classifier by minimizing the cross entropy (CE) loss $\mathcal{L}_{\text{CE}}$. The overall structure emulates a standard generative adversarial network (GAN) [36] with the REINFORCE augmenter functioning as the generator and the target classifier acting as the discriminator. IB loss [21] and several regularization losses, including consistency loss [37]

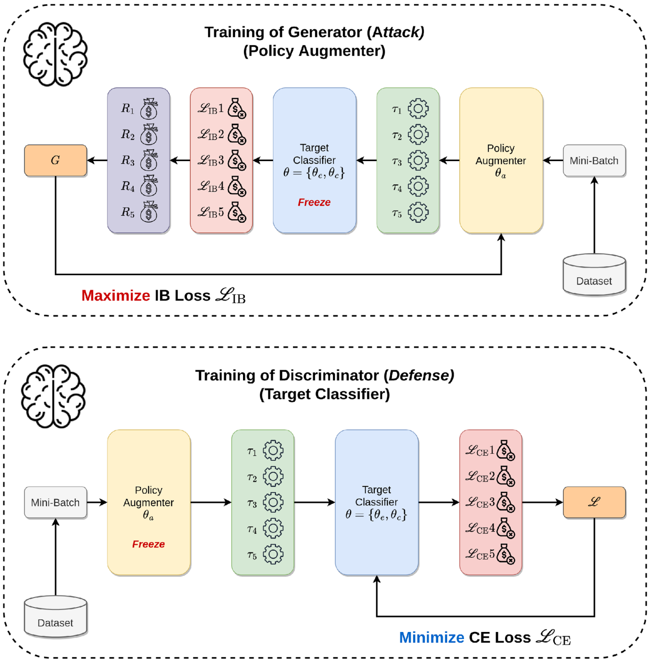

Training of Generator (Attack)
(Policy Augmenter)
G
R_1
R_2
R_3
R_4
R_5
$\mathscr{L}_{\mathrm{IB}}1$
$\mathscr{L}_{\mathrm{IB}}2$
$\mathscr{L}_{\mathrm{IB}}3$
$\mathscr{L}_{\mathrm{IB}}4$
$\mathscr{L}_{\mathrm{IB}}5$
Target Classifier $\theta = \{\theta_e, \theta_c\}$
Freeze
τ_1
τ_2
τ_3
τ_4
τ_5
Policy Augmenter θ_a
Mini-Batch
Dataset
Maximize IB Loss $\mathscr{L}_{\mathrm{IB}}$
Training of Discriminator (Defense)
(Target Classifier)
Mini-Batch
Dataset
Policy Augmenter θ_a
Freeze
τ_1
τ_2
τ_3
τ_4
τ_5
Target Classifier $\theta = \{\theta_e, \theta_c\}$
$\mathscr{L}_{\mathrm{CE}}1$
$\mathscr{L}_{\mathrm{CE}}2$
$\mathscr{L}_{\mathrm{CE}}3$
$\mathscr{L}_{\mathrm{CE}}4$
$\mathscr{L}_{\mathrm{CE}}5$
$\mathscr{L}$
Minimize CE Loss $\mathscr{L}_{\mathrm{CE}}$

Fig. 2. Adversarial training for generator (*attack*) and discriminator (*defense*). Training of REINFORCE augmenter for finding the worst-case augmentation strategy is illustrated at the top. The procedure for training the target classifier with the worst-case augmented data is shown at the bottom. τ_i is the final augmented data of the i^{th} data in mini-batch. G and $\mathcal{L}$ denote the average return and loss of a mini-batch, respectively.

and supervised contrastive loss [38] are introduced to maximize the adversarial data utilization.

A. Strategy Search for Worst-Case Augmentation

This subsection explains the proposed *attack* strategy to enable distinctive augmentation strategies for each data point, which cannot be achieved by applying greedy search.

1) Strategic Optimizer for Worst-Case Augmentation: To cope with the worst-case generalization in (4), it is essential to arrange dual representation and derive an effective reward setting that can meet the goal of fulfilling worst-case augmentation. For any distribution P_0 and any $\rho \geq 0$, solving (4) based on a loss function $\mathcal{L} : (\mathcal{X}, \mathcal{Y}) \times \Theta \to \mathbb{R}$ [39] is now handled by

$$\sup_{P}\{\mathbb{E}_P[\mathcal{L}(X, Y; \theta)] - \rho D_\theta(P_0, P)\} = \mathbb{E}_{\mathbf{x}_0 \sim P_0}[\ell_\rho(\mathbf{x}_0, \mathbf{y}_0; \theta)] \tag{5}$$

where $\ell_\rho(\mathbf{x}_0, \mathbf{y}_0; \theta)$ is seen as the robust surrogate loss [40]

$$\ell_\rho(\mathbf{x}_0, \mathbf{y}_0; \theta) \triangleq \sup_{\mathbf{x} \in \mathcal{X}}\{\mathcal{L}(\mathbf{x}, \mathbf{y}_0; \theta) - \rho c_\theta((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}, \mathbf{y}_0))\}. \tag{6}$$

We can satisfy $\nabla_\theta \ell_\rho(\mathbf{x}_0, \mathbf{y}_0; \theta) = \nabla_\theta \mathcal{L}(\mathbf{x}_\rho^\star, \mathbf{y}_0; \theta)$ by using the worst-case sample $\mathbf{x}_\rho^\star$ estimated by [41]

$$\mathbf{x}_\rho^\star = \arg\max_{\mathbf{x} \in \mathcal{X}}\{\mathcal{L}(\mathbf{x}, \mathbf{y}_0; \theta) - \rho c_\theta((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}, \mathbf{y}_0))\}. \tag{7}$$

In (6) and (7), the condition of label change $\mathbf{y}_o \neq \mathbf{y}$ after augmentation is not allowed due to the infinite cost in (2). In general, solving the penalty problem in (4) is equivalent to minimizing the robust surrogate loss ℓ_ρ defined in (6). Under suitable condition, the minimization of ℓ_ρ is identical to minimize the given loss $\mathcal{L}$ for the worst-case perturbed data $\mathbf{x}_\rho^\star$ of $\mathbf{x}_0$ using the current model parameter θ. In contrast to the prior works that searched the worst-case examples according to the maximization problem as shown by (7), this paper develops a novel solution that alternatively treats the worst-case data augmentation as a *reward maximization* problem by leveraging the benefits from reinforcement learning (RL).

The way to define the state representation is a crucial factor in enabling RL to feasibly search for an optimal augmentation strategy for each data point. In previous studies that utilized RL to identify optimal augmentation strategies in CV domain [16], [26], the states were defined based on the augmentation method applied to the entire dataset, such as the augmentation of images by a rotation of 15 degrees. As a consequence, the reward signal for RL was obtained in an expensive way, which was taken from the validation accuracy of a child model once the whole dataset was traversed by the selected augmentation action. In contrast, the current study defines the state as the embedding of each individual input sentence. Consequently, the reward signal can be obtained immediately after generating an augmented data. The initial state s_0 for ith original sentence $\mathbf{x}_0^i$ can be obtained from $s_0 = f(\mathbf{x}_0^i; \theta_e)$, and $s_t = f(\mathbf{x}_t^i; \theta_e)$ indicates the state of $\mathbf{x}_t^i$ where ith sentence has been transformed by REINFORCE augmenter for t times. REINFORCE [20] is a standard policy gradient method that maximizes the cumulative reward or return over a trajectory of length T. Return is defined as $G_t = \sum_{t'=0}^{T-1} \gamma^{t'} r_{t+t'+1}$. γ is a decay factor and r_t is the reward. REINFORCE algorithm is performed by updating the augmenter parameter θ_a via the following gradient ascent

$$\theta_a \leftarrow \theta_a + \eta_a \nabla_{\theta_a} \log \pi_{\theta_a}(a_t|s_t) G_t \tag{8}$$

where η_a is a learning rate and $\pi_{\theta_a}(a_t|s_t)$ is the policy probability of an action a_t. The same embedding parameter θ_e is used at each time step. By assigning individual action in each step, a series of stacked augmentations are performed to generate the worst-case adversarial examples which will be used to enhance the model robustness.

2) Reward for Worst-Case Reinforcement Learning: This study accomplishes the task of addressing the worst-case generalization by resolving (4), or equivalently by minimizing (6). This process is utilized to identify the optimal augmentation strategy to produce the worst-case sample $\mathbf{x}_\rho^\star$. This is done by following the objective of maximizing the subsequent reward

$$r_t = \mathcal{L}(\mathbf{x}_t, \mathbf{y}_0; \theta) - \rho c_\theta((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_t, \mathbf{y}_0)). \tag{9}$$

$\mathbf{x}_t$ is the augmented form of original sample $\mathbf{x}_0$ after transformation for t times. Since the RL agent is trained to maximize the reward, in every episode the agent learns how to arrange a sequence of actions to produce the augmented sample $\mathbf{x}_t$ that gradually gets close to the worst-case sample $\mathbf{x}_\rho^\star$ in (7).

However, it is necessary to refine the reward as defined in (9) to mitigate the influence of previously executed actions, thereby reflecting the true impact of action a_t. To cope with this issue, this study employs the reward function to capture the goal reaching the problem [42], which is characterized as the distance between the current state position and the goal position. In this

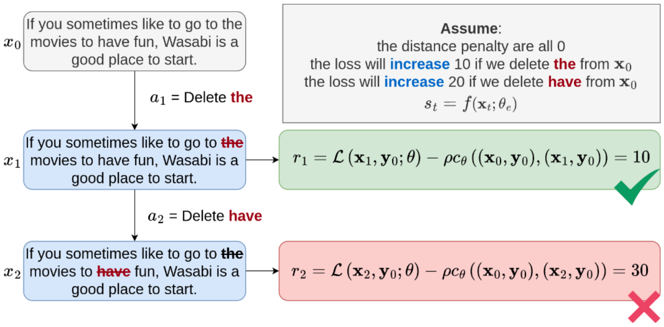

x_0
If you sometimes like to go to the movies to have fun, Wasabi is a good place to start.
a_1 = Delete the
x_1
If you sometimes like to go to ~~the~~ movies to have fun, Wasabi is a good place to start.
a_2 = Delete have
x_2
If you sometimes like to go to ~~the~~ movies to ~~have~~ fun, Wasabi is a good place to start.
Assume:
the distance penalty are all 0
the loss will increase 10 if we delete the from $\mathbf{x}_0$
the loss will increase 20 if we delete have from $\mathbf{x}_0$
$s_t = f(\mathbf{x}_t; \theta_e)$
$r_1 = \mathcal{L}\left(\mathbf{x}_1, \mathbf{y}_0; \theta\right) - \rho c_\theta\left((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_0)\right) = 10$
✓
$r_2 = \mathcal{L}\left(\mathbf{x}_2, \mathbf{y}_0; \theta\right) - \rho c_\theta\left((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_2, \mathbf{y}_0)\right) = 30$
✗

Fig. 3. Illustration of a failure that may occur if (9) is adopted as the reward function for RL. The obtained reward r_2 once a_2 was conducted cannot reflect the real impact of a_2 towards reaching the worst-case augmentation goal, because the influence of previous action a_1 is still considered. This failure can be avoided by using a new reward function defined in (10).

investigation, the worst-case augmentation data $\mathbf{x}_\rho^\star$ is considered as the goal. Consequently, the reward can be represented as the distance between the current observation and the goal observation, expressed as $r(s_t, g_t, a_t, s_{t+1}) = -\|s_t + g_t - s_{t+1}\|_2$, where g_t is the ultimate goal the agent aspires to attain, and s_t is the current observation. By implementing this reward function, the efficacy of a_t can be evaluated from how much the distance is shortened to reach the final goal g_t from the previous observation s_{t-1} due to conducting an action a_t. By integrating this reward scenario into (9), a new reward function that takes into account the influence from the previous actions can be defined as

$$\begin{aligned} r_t = {} & \mathcal{L}(\mathbf{x}_t, \mathbf{y}_0; \theta) - \rho c_\theta((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_t, \mathbf{y}_0)) \\ & - \mathcal{L}(\mathbf{x}_{t-1}, \mathbf{y}_0; \theta) + \rho c_\theta((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_{t-1}, \mathbf{y}_0)). \end{aligned} \quad (10)$$

The subtraction with the previously obtained reward by the agent is intended to reveal the pure reward that the agent achieves after conducting the latest action a_t. Considering the sequence of text augmentations illustrated in Fig. 3, if we define reward as (9), we will obtain $r_2 = 30$ which does not represent the true reward after conducting a_2. This is because (9) directly compares $\mathbf{x}_2$ (which contains the impact of a_1) with $\mathbf{x}_0$. Based on the new reward in (10), the true reward which is 20 can be obtained.

Importantly, the preservation of label $\mathbf{y}_0$ is required to avoid infinite loss in (2). Practically, the randomness of behavior policy can be suppressed by only considering the distance penalty but without taking the term $\infty \cdot \mathbb{1}\{\mathbf{y}_0 \neq \mathbf{y}\}$ into account. However, this reward definition is still insufficient to prevent the extreme case of augmentation such as deleting all tokens of a sentence until leaving only one comma when the original sentence is very short. Therefore, this work adopts a cosine similarity constraint to ensure the property of label-preserving during augmentation procedure

$$r_t^\star = \begin{cases} 0, & \text{if } \cos(s_0, s_t) < \alpha \\ r_t, & \text{else.} \end{cases} \quad (11)$$

where $r_t^\star$ is the final reward after considering the constraint over the states or semantic embeddings. If the cosine similarity between the embeddings of the original sentence s_0 and the augmented sentence s_t falls below a predefined threshold α, this scenario is considered as a label-altering augmentation, thereby nullifying the reward. Consequently, the augmentation episode for the sentence is terminated. This setting encourages the policy to maximize the reward by seeking the most challenging augmentation strategy within a suitable similarity threshold. Due to the implementation of the constraint in (11), this work eliminates the need for evaluators to assess the validity of the augmented data generated by the model.

3) Information Bottleneck Reward Maximization: The target loss $\mathcal{L}$ is maximized for exploration in worst-case augmentation where the cross entropy loss $\mathcal{L}_{\text{CE}}$ is a common loss function for classification task. However, a meaningful loss defined in $\mathcal{L} : (\mathcal{X}, \mathcal{Y}) \times \Theta \rightarrow \mathbb{R}$ can be consolidated to improve the worst-case generalization. This paper presents an informative and strategic augmentation where a new loss function is formed and optimized through the IB reward. Accordingly, the reward r_t in (10) is refined to maximize the loss in (6) for worst-case generalization in accordance with IB loss $\mathcal{L}_{\text{IB}}(X, Y; \theta) = I(X; Z) - I(Y; Z)$ where Z is the compressed embeddings from inputs $X = \{\mathbf{x}^i\}_{i=1}^N$, and this Z is used for prediction of targets $Y = \{\mathbf{y}^i\}_{i=1}^N$. An adversarial augmentation is performed by maximizing $\mathcal{L}_{\text{IB}}$ during *attack* stage to find the worst-case augmentation strategy and minimizing $\mathcal{L}_{\text{IB}}$ to estimate the target classifier with the compressed latent space Z in the subsequent *defense* stage. Typically, a target model with the compressed Z is hard to generalize for out-of-domain data due to the compression loss. In order to overcome this issue, we train a policy augmenter by leveraging the reformulated IB loss where each of mutual information (MI) components, $I(Y; Z)$ and $I(X; Z)$, is individually estimated [43]. First, MI between target label and latent compression $I(Y; Z)$ can be revealed for prediction risk where the cross entropy (CE) loss

$$\mathcal{L}_{\text{CE}}(X, Y; \theta) = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_c^i \log p_\theta(y_c^i | \mathbf{x}^i) \quad (12)$$

is adopted to reflect $-I(Y; Z)$. Here, $\mathbf{y}^i = \{y_c^i\}$ is a one-hot output vector with C classes. Meanwhile, MI between input data and compressed embedding $I(X; Z)$ can be approximated by a simple entropy based on the data processing inequality $I(X; Z) \geq I(X; \widehat{Y})$ [21], [44] where $\widehat{Y}$ is the predicted label. Furthermore, because of the property of label preserving in worst-case augmentation, the predicted label $\widehat{Y}$ from an input X is fixed which results in conditional entropy $H(\widehat{Y}|X) = 0$. The inequality is used to find lower bound of $I(X; Z)$ as

$$I(X; Z) \geq I(X; \widehat{Y}) = H(\widehat{Y}) - H(\widehat{Y}|X) = H(\widehat{Y}) \quad (13)$$

where $H(\widehat{Y})$ can be directly calculated from the prediction of model. Alternatively, the *lower bound* of IB loss

$$\widetilde{\mathcal{L}}_{\text{IB}}(X, Y; \theta) = \mathcal{L}_{\text{CE}}(X, Y; \theta) + \beta H(\widehat{Y}) \quad (14)$$

is maximized to find the worst-case augmentation with a hyperparameter β. Redefining r_t in (10) by replacing $\mathcal{L}(\mathbf{x}_t, \mathbf{y}_0; \theta)$ and $\mathcal{L}(\mathbf{x}_{t-1}, \mathbf{y}_0; \theta)$ with the corresponding IB losses $\widetilde{\mathcal{L}}_{\text{IB}}(\mathbf{x}_t, \mathbf{y}_0; \theta)$ and $\widetilde{\mathcal{L}}_{\text{IB}}(\mathbf{x}_{t-1}, \mathbf{y}_0; \theta)$ in (14) is similar to encourage more stochastic action due to the entropy maximization, which promotes more policy exploration for finding as many as possible worst-case augmentation scenarios. Therefore, considering (10) and (11), the novelty of this work is to introduce the reward function derived from the information bottleneck (IB) theory. The generalization and robustness are improved via *uncertainty modeling* [45] in IB while the transportation cost c_θ is minimized to constrain the distance to original data.

B. Adversarial Augmentation and Classification

This study works on an adaptive and adversarial data augmentation where a policy augmenter is jointly trained with a target classifier according to an informative and regularized objective via reinforcement learning. Adversarial attack and defense are tightly coupled and collaboratively performed [46]. Consistency learning and contrastive learning are implemented for regularization in *defense* stage.

1) Adversarial Learning for Attack and Defense: Importantly, the target classifier and the policy augmenter are jointly trained to improve the generalization of a model and assure the robustness to unseen noisy inputs. A kind of adversarial learning [47] is performed to minimize the upper bound of IB loss $\widetilde{\mathcal{L}}_{\text{IB}}$ for text classifier (*defense*) and simultaneously maximize $\widetilde{\mathcal{L}}_{\text{IB}}$ for policy augmenter (*attack*). Policy augmenter and text classifier are associated with the generator and the discriminator in adversarial learning based on GAN [36], [48], [49], respectively. The augmentation strategy is designed to explore the search space in a way of finding the *most diverse augmentation* with the *least classification inconsistency* between augmented and original sentences. Conceptually, such an adversarial learning for augmentation and classification is viewed as finding the worst-case augmenter to *attack* for unknown data regions and at the same time estimating the best classifier to *defend* for the most consistent performance. Following the perspectives of *exploration* and *exploitation* in RL, this study builds a robust learning machine where the augmenter explores as much as possible the unknown states in data space and the classifier exploits its best capability by utilizing the integrated dataset. In order to exploit the benefit of those generated data, additional model regularization can be introduced to improve the performance of target classifier. Regularization is imposed to further enhance the robustness of a target model. In general, there are two types of regularization for data exploitation. The first one is the *local* regularization that is implemented by considering the relation between individual data points with the corresponding augmentation. The second one is the *global* regularization which is performed by considering the relation among individual data points in a mini-batch. Following this direction, this study presents two regularization terms to carry out a new target classifier along with the IB objective $\widetilde{\mathcal{L}}_{\text{IB}}$ which is maximized for text augmenter and minimized for target classifier.

2) Regularization for Defense and Implementation: For further exploiting the benefit of the augmented data, this paper introduces additional regularization for classifier by encouraging the *consistency* between original and augmented data as well as leveraging the *contrastive* information among different classes. The *consistency* and *constrastive* objectives are related to the *local* and *global* regularization, respectively. Consistency loss is designed to exploit the augmented data by strengthening the target model to aware for a condition that each data point and its worst-case augmentation have close semantic meaning. The prediction $\mathbf{y}^i$ of the ith data point $\mathbf{x}^i$ is forced to be consistent with the corresponding augmented data $(\mathbf{x}^i)^\star$. The consistency (CT) loss is therefore defined as

$$\mathcal{L}_{\text{CT}} = \sum_{i=1}^{N} \text{JS}(p_\theta(\mathbf{y}^i|\mathbf{x}^i), p_\theta(\mathbf{y}^i|(\mathbf{x}^i)^\star)) \tag{15}$$

where the Jensen-Shannon (JS) divergence is measured by using the conditional likelihood for classification $p_\theta(\mathbf{y}^i|\mathbf{x}^i)$. We use JS divergence because it is bounded and more stable in comparison with Kullback-Leibler (KL) divergence [18], [50]. On the other hand, a supervised contrastive (SC) loss [38] is leveraged to impose the global regularization where the generation of worst-case augmented data is built upon the awareness of class information. Contrastive learning is fulfilled by leveraging the class information which regularizes the model to learn among different classes. Accordingly, SC loss is introduced and calculated by

$$\mathcal{L}_{\text{SC}} = -\sum_{i=1}^{N} \frac{1}{N_{y^i}-1} \sum_{j=1}^{N} \mathbb{1}_{i\neq j} \mathbb{1}_{\mathbf{y}^i=\mathbf{y}^j} \times \log \frac{\exp((f(\mathbf{x}^i;\theta_e))^\top f(\mathbf{x}^j;\theta_e)/\epsilon)}{\sum_{k=1}^{N} \mathbb{1}_{i\neq k} \exp((f(\mathbf{x}^i;\theta_e))^\top f(\mathbf{x}^k;\theta_e)/\epsilon)} \tag{16}$$

where N_{y^i} denotes the total number of examples in a mini-batch that have the same label $\mathbf{y}^i$, and ϵ is a temperature parameter that controls the separation of classes. This contrastive loss is measured via the contrastive information using different samples $\mathbf{x}^j$ and $\mathbf{x}^k$ in numerator and denominator, respectively, by considering the labels $\mathbf{y}^i$ and $\mathbf{y}^j$ of different samples $\mathbf{x}^i$ and $\mathbf{x}^j$. The regularized loss $\widetilde{\mathcal{L}}$ for adversarial augmenter and classifier is constructed by

$$\widetilde{\mathcal{L}}((X, X^\star), Y; \theta) = \widetilde{\mathcal{L}}_{\text{IB}} + \lambda_{r_1}\mathcal{L}_{\text{CT}} + \lambda_{r_2}\mathcal{L}_{\text{SC}} \tag{17}$$

where $X^\star = \{(\mathbf{x}^i)^\star\}_{i=1}^N$ and λ_{r_1} and λ_{r_2} denote the regularization parameters for adjusting the importance of local and global regularization, respectively.

The whole algorithm of the proposed worst-case augmenter and classifier (WAC) is implemented by Algorithm 1 where n, N_c, and T, denote the numbers of performing attack and defense, training steps for classifier, and augmenting steps for each data point, respectively. There are three phases for training the target classifier and policy augmenter based on worst-case augmentation. The first phase is to train the target model by only utilizing the set of clean data. After obtaining the well-trained target model, the policy augmenter is trained in the second phase to deal with the weakness of target classifier by augmenting adversarial examples. Lastly, in order to improve the model robustness and tackle the generalization to unseen noisy inputs, the target model is optimized by using the worst-case examples which are synthesized by policy augmenter from the second phase. Since the proposed method follows the adversarial learning procedure, the model can be further improved by applying attack in the second phase and defense in the third phase for several times.

Algorithm 1: Training for the Worst-Case Augmenter and Classifier (WAC).

```
Require: Training set $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}$, pars. of encoder,
         classifier $\theta = \{\theta_e, \theta_c\}$, augmenter $\theta_a$,
         hyperparams. $\{\eta_a, \eta_c, \alpha, \beta, \gamma, \rho, \epsilon, \lambda_{r_1}, \lambda_{r_2}\}$
1st phase: training classifier
for $j = 1, \ldots, N_c$ do
    sample a mini-batch $\{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^m$ from $\mathcal{D}$
    calculate $\mathcal{L}_{\text{CE}}$ via Eq. (12)
    $g_\theta \leftarrow \nabla \frac{1}{m} \sum_{i=1}^m \mathcal{L}_{\text{CE}}(\mathbf{x}^i, \mathbf{y}^i; \theta)$
    $\theta \leftarrow \eta_c \cdot \text{AdamW}(\theta, g_\theta)$
end
for $k = 0, \ldots, n-1$ do
    2nd phase (attack): training augmenter
    for i=1, ..., N do
        input $i^{\text{th}}$ original data $\mathbf{x}^i$
        calculate state $s_0 = f(\mathbf{x}^i; \theta_e)$
        for $t = 0, \ldots, T-1$ do
            find action $a_t$ and augmented data $(\mathbf{x}^i)^\star$
            calculate state $s_t = f((\mathbf{x}^i)^\star; \theta_e)$
            calculate $r_t$ via Eqs. (10)(14)
            store $s_t$, $a_t$, $r_t$
        end
        $\{s_0, a_0, r_0, ..., s_{T-1}, a_{T-1}, r_{T-1}\} \sim \pi_{\theta_a}$
        $G_t = \sum_{t'=0}^{T-1} \gamma^{t'} r_{t+t'+1}$
        $g_{\theta_a} \leftarrow -\nabla_{\theta_a} \log \pi_{\theta_a}(a_t|s_t) G_t$ via Eq. (8)
        $\theta_a \leftarrow \text{Adam}(\theta_a, \eta_a, g_{\theta_a})$
    end
    3rd phase (defense): optimizing classifier
    for $j = 1, \ldots, N_c$ do
        sample a mini-batch $\{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^m$ from $\mathcal{D}$
        estimate $\{(\mathbf{x}^i)^\star\}_{i=1}^m$ via augmenter
        calculate $\widetilde{\mathcal{L}}$ via Eqs (17)(14)(15)(16)
        $g_\theta \leftarrow \nabla \frac{1}{m} \sum_{i=1}^m \widetilde{\mathcal{L}}((\mathbf{x}^i, (\mathbf{x}^i)^\star), \mathbf{y}^i; \theta)$
        $\theta \leftarrow \text{AdamW}(\theta, \eta_c, g_\theta)$
    end
end
```

Adam [51] and AdamW [52] are used as the optimizers to train policy augmenter and target classifier, respectively.

V. Experiments

A series of experiments were conducted to analyze the properties of the proposed method through evaluation of different tasks for text classification.

A. Experimental Settings

In this study, text augmentation was performed under five discrete actions in the action set given as $A = \{a_t\} = \{\text{RD}, \text{RS}, \text{SR}, \text{SI}, \text{STOP}\}$ which included random deletion, random swap, synonym replacement, synonym insertion and stop augmentation with the examples shown in Table I. Stop action means that agent decides to stop augmenting the sentence. A number of datasets were adopted in the evaluation and detailed in what follows.

TABLE I
Examples of Augmented and Original Sentences Via Five Different Augmentation Actions

Actions	Aug. and Orig. Sentences
0: **RD** (rand delete)	~~Sparse~~ only curiously compelling
1: **RS** (rand swap)	**compelling** only curiously **Sparse**
2: **SR** (syn replace)	Sparse only **oddly** compelling
3: **SI** (syn insert)	Sparse only curiously **oddly** compelling
4: **Stop**	Sparse only curiously compelling

Stop means the agent stops the augmentation, namely the original sentence.

TABLE II
Summary of Number of Classes (Class), Average Length of Sentences (Len), Vocabulary (Voc) Size and Others in Datasets

Dataset	Class	Len	Train Size	Val Size	Test Size	Voc Size
SST-2	2	19	6920	872	1821	14838
SST-5	5	18	8544	1101	2210	16262
CR	2	19	2548	283	944	5046
MPQA	2	3	7159	795	2652	6246
SUBJ	2	23	6750	750	2500	21323
TREC-6	6	10	5357	595	500	9592

1) Experimental Datasets: There were six tasks which were collected to investigate different methods for text classification. Summary of different statistics in individual datasets are provided in Table II.

1) Stanford Sentiment Treebank (SST) [53] is a sentiment classification dataset. Most of sentiment contents were collected from the movie reviews in Rotten Tomatoes. This work adopted both the binary classification in SST-2 and the fine-grained classification in SST-5.
2) Customer Review (CR) dataset [54] contains customer reviews of five different electronics products. In the evaluation, only the text features were leveraged to predict the sentiment.
3) Multi-Perspective Question Answering (MPQA) [55] is an opinion corpus which contains news articles from a wide variety of news sources manually annotated by opinions and other private states. In the evaluation, MPQA version 1.0 was used to do polarity classification.
4) SUBJ dataset is comprised of movie-review documents, which are categorized based on their overall sentiment polarity and subjectivity status.
5) TREC-6 [56] is a dataset consisting of open-domain and fact-based questions divided into six broad semantic categories.

2) Comparison From Different Perspectives: In contrast to previous studies that evaluated their methods on less advanced models such as LSTM [57] and BERT [58], this paper demonstrates the efficacy of the proposed strategic and adversarial training on the robust and high-performance RoBERTa model [59]. This approach provided the compelling evidence. If the proposed adversarial training yields performance nearly identical to the original RoBERTa, it would imply that the proposed method does not offer significant benefit. Three contextual ADA methods including DeepWordBug (DWB) [8], PWWS [6] and TextBugger [7] were selected as the baselines as all of them show competitive performances. As the proposed worst-case augmenter and classifier (WAC) also conducts the contextual

TABLE III
COMPARISON OVER DIFFERENT CONTEXTUAL ADVERSARIAL DATA AUGMENTATION METHODS FROM DIFFERENT PERSPECTIVES

Method	Goal	Constraint	Transformation	Search
DWB [8]	untarget (cross entropy)	semantic (Levenshtein distance)	char-based (delete, swap, replace, insert)	Greedy-WIR
PWWS [6]	untarget (cross entropy)	–	word-based (replace)	Greedy-WIR
TextBugger [7]	untarget (cross entropy)	overlap (USE embedding [60])	char-based (delete, swap, replace, insert)	Greedy-WIR
WAC	untarget (information bottleneck)	semantic (obtained reward)	word-based & stacking (delete, swap, replace, insert)	reinforcement learning

TABLE IV
HYPERPARAMETER SETTINGS FOR CLASSIFIER TRAINING IN SIX CLASSIFICATION TASKS

Dataset	TS	WS	η_a	η_c	α	β	λ_{r_1}	λ_{r_2}
SST-2	1300	80	1e-4	1e-5	0.8	0.9	1	0.5
SST-5	1600	96	1e-4	1e-5	0.8	0.9	1	0.5
CR	500	30	1e-4	1e-5	0.8	0.9	1	0.5
MPQA	1260	75	1e-4	1e-5	0.8	0.9	1	0.5
Sunj	1260	75	1e-4	1e-5	0.8	0.9	1	0.5
TREC-6	920	55	1e-4	1e-5	0.8	0.9	1	0.5

WS stands for the number of warmup steps in the training steps (TS).

TABLE V
AN ORIGINAL SENTENCE AND ITS NOISY SENTENCES GENERATED BY USING 10 DIFFERENT NOISES

Original sent.	**Occasionally melodramatic, it's also extremely effective.**
BackTrans-De	Occasionally melodramatic, it is also extremely effective.
BackTrans-Ru	At first glance, it is effective, but it is also extremely effective.
BackTrans-Zh	Occasionally dramatic, it's also very effective.
Charswap	Occasionally melodramatic, it's alMo extremely effective
Checklist	Occasionally melodramatic, it is also extremely effective
Clare	Occasionally melodramatic, it's Ġproving extremely effective.
EDA	Occasionally also, it's melodramatic extremely effective.
Embedding	Occasionally melodramatic, it's also insanely effective.
Spell	Occasunaily melodramatic, i't ' s malso extremely effective.
Stack EDA	melodramatic, also it's besides extremely.

The differences between original and noisy sentences are displayed in red.

augmentation, Table III shows the differences between WAC and the contextual ADA baselines in terms of *goal*, *constraint*, *transformation*, and *search*. Firstly, the *goal* of all methods is basically *untarget* as the attacker tries to perturb the original classification output *without* targeting a specific class. However, the proposed WAC is optimized based on the information bottleneck objective which can model the uncertainty, as shown in (14). Secondly, the *constraint* pertains to the rules for valid transformation. While TextBugger uses an overlap constraint to determine the validity of a perturbation based on character-level analysis, WAC maintains the semantic closeness in data augmentation through word-level analysis. It does not necessitate the constraints of applying the USE embedding [60] or a predetermined Levenshtein distance to assess the generated examples. Instead, the validity of augmented data is ascertained via the obtained reward. Thirdly, the *transformation* is the action to transform the data. Different from the other methods, the transformation in WAC is flexible and adaptive as it is designed by *stacking* the actions. Lastly, the *search* is a method to explore the space of potential transformation. WAC utilizes RL to reduce the computation cost without the heavy overhead like in the greedy search with word importance ranking (WIR). In addition to the contextual augmentation methods, a strong representational augmentation, called the adversarial word dilution (AWD) [34], is also employed as a baseline method. AWD augments the textual data by diluting the embedding of highly positive words with the unknown-word embedding.

3) Hyperparameter Settings: Since PWWS [6], DWB [8], and TextBugger [7] were not originally applied to RoBERTa, their hyperparameter settings were set by referring to [61] and original RoBERTa [59]. In order to make a fair comparison among adversarial methods, the total number of attacks and defenses were set identically, and the proposed method only traversed the whole dataset once. Table IV shows the hyperparameters for individual datasets in the experiments which were empirically selected according to the validation accuracy obtained in each setting. The other hyperparameters such as temperature ϵ, decay factor γ and relaxation penalty ρ were fixed as 0.07, 0.99 and 1, respectively.

4) Construction of Noisy Test Set: The system robustness was evaluated on various unseen noisy data. The unseen noisy datasets were constructed to simulate the potential disturbances that could arise in real-world applications. These disturbances encompassed the character, word, and sentence-level noises. A total of ten augmentation methods were utilized in the construction of the noisy test set. These methods were SEDA, EDA [29], Embed, Clare [62], Checklist [63], Charswap, and Back-translation [27] from three languages, namely German (De), Russian (Ru), and Chinese (Zh), and Spell. In further details, EDA incorporates four distinct augmentation operations: delete, swap, replace, and insert. SEDA is viewed as an advanced iteration of EDA. It stacks several augmentation operations provided in EDA. Embed, on the other hand, augments an input by substituting its words with synonyms in the word embedding space, leveraging Glove embedding. Clare, constructed on a pre-trained masked language model, modifies the inputs contextually. It includes three contextualized perturbations: replace, insert, and merge, which allow for the generation of outputs with varied lengths. Checklist perturbs the words using transformation methods provided by CheckList INV testing [63], enabling a combination of several features such as name replacement, location replacement, and number alteration. Charswap augments the words by swapping characters with the other characters. Lastly, Spell modifies words based on a spelling mistake dictionary. Table V shows the examples of the noisy sentences generated by various augmentation methods. Although these methods are popularly used, the augmented sentences are hard to understand.

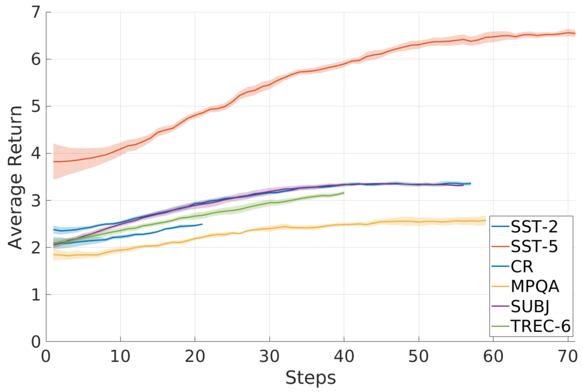


Fig. 4. Average return obtained by augmenter model after every mini-batch training on five different seeds in six different datasets.

TABLE VI
AVERAGE ACCURACIES (%) OVER CLEAN AND TEN NOISY TEST SETS

Method	**SST-2**	**SST-5**	**CR**	**MPQA**	**SUBJ**	**TREC-6**	**Avg.**
RoBERTa [59]	90.85	52.48	90.40	83.45	93.52	87.28	83.00
DWB [8]	90.04	52.25	88.16	82.00	94.47	89.49	82.74
PWWS [6]	90.75	52.33	90.93	83.63	94.18	88.87	83.45
TextBugger [7]	89.69	52.42	91.10	83.50	**94.65**	88.97	83.39
AWD [34]	90.02	52.63	90.06	83.55	94.03	89.29	83.26
WAC	**91.86**	**53.65**	**91.50**	**84.44**	94.49	**91.42**	**84.56**

The results of using individual models and datasets are shown.

B. Results on Robustness Evaluation

Before evaluating model robustness on the noisy test set, Fig. 4 depicts the learning curve of the augmenter training in each dataset which shows that the augmenter model successfully learned appropriate policy at the end of the training step, indicated by high average return, G. The policy augmenter received the largest average return in the SST-5 dataset as it is the hardest task considering 5 class labels and long sentences with large vocabulary size. As a consequence, the classifier could be easily fooled by the augmenter which resulted in high $\widetilde{\mathcal{L}}_{\text{IB}}$ while still maintaining reasonable cosine similarity. Next, Table VI presents a comparative analysis of text classification accuracies, averaged over clean and ten constructed noisy test set. The findings from six tasks, namely SST-2, SST-5, CR, MPQA, SUBJ, and TREC-6, are reported. RoBERTa, listed at the top of the table, was trained without the use of adversarial training. The highest accuracy in each task is denoted in bold.

Several findings can be drawn from Table VI. Primarily, in comparison to the baselines, only the proposed WAC consistently outperformed RoBERTa across the six tasks. Upon averaging the results from the six distinct classification tasks, it was found that all of the baseline methods performed nearly equivalently to the standard RoBERTa training. Meanwhile, WAC was able to enhance the accuracy by over 1.5%, highlighting the advantages of employing WAC. In more details, for the most challenging task, sentiment classification in SST-5 dataset, the other contextual adversarial augmentation using DWB, PWWS and TextBugger exhibited inferior performance compared to RoBERTa, trained by only using the clean dataset. Meanwhile, AWD which applied the representational augmentation by perturbing the embedding space only demonstrated very small improvement. Next, in the TREC-6 dataset, all methods showed remarkable performance. This is likely because the true semantic meaning in the question classification task can be easily captured even when the sentences are perturbed. In the SUBJ classification task, the room of improvement is relatively limited due to the fact that RoBERTa has already shown a convincing performance. Additionally, SUBJ dataset has relatively large vocabulary size compared to the other dataset which makes the generation of adversarial data become very challenging.

Table VII presents the average accuracies of individual models under various types of perturbations, with the results averaged across six datasets. The proposed WAC model exhibited consistent improvement over all baseline models in clean and ten different noisy conditions. Meanwhile, the other adversarial methods such as DWB, PWWS, TextBugger, and AWD reported the accuracies that are comparable to the standard RoBERTa. In an extreme scenario where RoBERTa exhibited bad performance, for instance in the noisy test sets constructed by SEDA, EDA, Spell and back-translation from Chinese, the benefit of using WAC became more evident. Notably, WAC maintained the original performance on the clean dataset, a performance that the other adversarial models were unable to accomplish. As the noise level in the perturbations increased, the benefits of WAC became more significant.

C. Results on Behavior Analysis

To analyze the behavior of the policy augmenter in WAC, some augmented samples from SST-5 and TREC-6 tasks were utilized as the examples, as illustrated in Table VIII. It was observed from these randomly selected examples that the learned policy tends to function more effectively in longer sentences compared to shorter ones, as indicated by the cumulative obtained reward. This means that the policy can easily fool the target model when given by long sentences. The generated examples typically provide the target classifier with the ability to manage the perturbations in text classification task. However, for the shorter sentences, the policy behavior for introducing perturbations become limited, resulting in the minimal actions. This leads to the augmented data being easily distinguished by the target classifier.

Further analysis regarding the policy behavior is provided by Table IX, which shows the action distribution of the policy in terms of occupation probability across different attack-defense rounds in SST-5 and TREC-6. Intuitively, once the target model learns to defend against the worst-case samples generated by the policy in the defense round, the policy should adopt different augmentation strategies to deceive the target model in the subsequent attack round. Consequently, the occupation probabilities of the actions during different attack rounds in training vary. For instance, in SST-5, it is evident that the policy altered its strategy in rounds 1, 4, and 5. In contrast, for TREC-6, the policy exhibited distinct behavior in individual rounds. This observation demonstrates that the proposed method has the capability to adaptively find an appropriate strategy to improve the target model by deceiving it during training.

TABLE VII
AVERAGE ACCURACIES (%) OVER SIX DATASETS

Method	**Clean**	**SEDA**	**EDA**	**Embed**	**Clare**	**Checklist**	**Charswap**	**De**	**Ru**	**Zh**	**Spell**	**Avg.**
RoBERTa [59]	88.00	74.97	84.31	86.08	85.03	87.76	83.46	84.58	83.04	80.45	80.28	83.45
DWB [8]	88.48	74.52	84.75	86.57	84.54	88.28	81.97	84.56	83.41	80.94	77.79	83.26
PWWS [6]	87.95	76.11	84.86	86.33	85.47	87.70	84.06	84.35	83.03	81.11	81.47	83.86
TextBugger [7]	88.23	75.97	84.54	86.57	85.14	87.96	84.07	84.73	83.59	81.43	79.86	83.83
AWD [34]	87.95	74.74	84.11	85.45	84.71	87.68	82.83	84.61	83.29	81.47	79.05	83.26
WAC	**88.69**	**79.75**	**86.43**	**86.76**	**86.43**	**88.39**	**84.50**	**85.19**	**83.96**	**82.08**	**83.13**	**85.03**

The results of using individual models and noisy perturbations are shown.

TABLE VIII
ILLUSTRATION FOR THE STACKED DATA AUGMENTATION IN WAC WITH FIVE ACTIONS (0: RANDOM DELETE, 1: RANDOM SWAP, 2: SYNONYM REPLACE, 3: SYNONYM INSERT, 4: STOP)

Original Sentence (SST-5)	**Augmented Sentence**	**Ep. Action**	**Ep. Reward**
We don't even like their characters	We ilk don't regular like their characters wish.	3, 2, 3, 2, x	1.93
Rarely has skin looked as beautiful, desirable, even delectable, as it does in Trouble Every Day	Rarely has looked as beautiful, desirable, even delectable it does in, Every Day.	0, 1, 0, 4	5.35
Shyamalan takes a potentially trite and overused concept -LRB- aliens come to Earth -RRB- and infuses it into a rustic, realistic, and altogether creepy tale of hidden invasion.	Shyamalan takes a potentially get trite and overdrive concept -LRB- alienate creepy crawly and to Earth -RRB- come infuses it into a rustic, realistic creepy crawly, and altogether inculcate creepy tale of hidden invasion.	3, 2, 2, 3, 1, 3	5.84

Original Sentence (TREC-6)	**Augmented Sentence**	**Ep. Action**	**Ep. Reward**
urging	urging	4	0
bitter deadlock	deadlock dead stalemate bitter closing virulent	3, 3, 2, 1, 1, 3	3.55
rising chorus of domestic and international outrage	and chorus of domestic domestic help rising international outrage	1, 1, 1, 1, 3, 1	3.15

"x" denotes the failed action due to losing of original semantic meaning, indicated by the condition cos(s_0, s_t) < α. The action order and the obtained cumulative reward in an episode (ep.) are shown. (top) policy behavior in sst-5 dataset. (bottom) policy behavior in trec-6 dataset.

TABLE IX
ILLUSTRATION OF OCCUPATION PROBABILITIES (%) OF DIFFERENT ACTIONS FOR THE AGENT

Operation	**Attack 1**	**Attack 2**	**Attack 3**	**Attack 4**	**Attack 5**
Delete	14.82	26.51	16.65	**42.54**	18.37
Swap	9.26	9.97	9.92	15.70	8.55
Replace	36.31	13.01	26.83	25.16	**37.30**
Insert	**39.61**	**50.51**	**46.61**	16.60	35.78

Operation	**Attack 1**	**Attack 2**	**Attack 3**	**Attack 4**	**Attack 5**
Delete	26.85	27.16	**32.44**	18.22	30.26
Swap	**27.33**	21.21	23.15	24.25	13.06
Replace	25.60	16.76	14.17	25.41	13.06
Insert	20.23	**34.87**	30.24	**32.12**	**43.61**

Attack 1, 2, 3, 4 indicates the index of attack in attack-defense. Attack 2 means that the target model is attacked by the second time, after defending one time. (top) action distribution in SST-5 dataset. (bottom) action distribution in TREC-6 dataset.

TABLE X
THE COSINE SIMILARITY BETWEEN ORIGINAL AND AUGMENTED DATA IN DIFFERENT ATTACK-DEFENSE ROUNDS

Operation	**Attack 1**	**Attack 2**	**Attack 3**	**Attack 4**	**Attack 5**
Delete	0.8816	0.9289	0.9297	0.9354	**0.9374**
Swap	0.8888	0.9403	0.9438	0.9335	**0.9451**
Replace	0.9129	0.9412	0.9470	0.9408	**0.9517**
Insert	0.9349	0.9529	0.9542	0.9592	**0.9631**

Operation	**Attack 1**	**Attack 2**	**Attack 3**	**Attack 4**	**Attack 5**
Delete	0.8741	**0.9464**	0.9321	0.9457	0.9455
Swap	0.8950	0.9421	0.9693	0.9725	**0.9813**
Replace	0.9181	0.9467	**0.9743**	0.9578	0.9674
Insert	0.9262	0.9607	0.9773	**0.9834**	0.9771

(Top) cosine similarity in SST-5 dataset. (bottom) cosine similarity in TREC-6 dataset.

Lastly, an analysis of target model was conducted, predicated on the initial assumption that, given a perfect target model encoder and perfect label-preserving transformation, the cosine similarity between the embedding of the original sentence and that of the augmented sentence should be 1, regardless of the number of transformations performed by the label-preserving augmentation operations. Table X shows a phenomenon existed in the target classifier given by adversarial data and original data in different attack-defense rounds in which the cosine similarity between augmented and original data is getting closer to 1 as the attack-defense round increases. This evidence demonstrates that the encoder in target classifier treated the embedding of the augmented sentences identically to the original sentences as both of target classifier encoder and policy augmenter getting closer to the optimum property which follows the predefined assumption.

D. Human Evaluation on Augmented Data

The evaluation was extended to include a human assessment for the quality of the generated sentences by various data augmentation methods. This evaluation process involved randomly sampling 100 instances from the training set. The quality of the augmented data was measured in terms of fluency and semantic similarity, focusing on assessing the readability and whether the augmented data preserved essential information from the original sentences. Four annotators provided the scores ranging from 0 to 5. The averaged scores are reported in Table XII. The generated sentences by using the proposed WAC consisting of a

TABLE XI
Average Accuracies (%) Over Ten Noise Perturbations From WAC With Different Reward Functions (Top) and Different Defense Strategies (Bottom)

Method	SST-2	SST-5	CR	MPQA	SUBJ	TREC-6	Avg.
WAC	91.86	53.65	91.50	84.44	94.49	91.42	84.56
WAC without IB	91.39 (↓ 0.47)	53.15 (↓ 0.50)	91.20 (↓ 0.30)	84.00 (↓ 0.44)	94.32 (↓ 0.17)	91.05 (↓ 0.37)	84.18 (↓ 0.38)

Method	SST-2	SST-5	CR	MPQA	SUBJ	TREC-6	Avg.
WAC	91.86	53.65	91.50	84.44	94.49	91.42	84.56
WAC without CT	91.33 (↓ 0.53)	53.07 (↓ 0.58)	91.26 (↓ 0.24)	84.13 (↓ 0.31)	94.18 (↓ 0.31)	90.72 (↓ 0.70)	84.11 (↓ 0.45)
WAC without SC	91.58 (↓ 0.28)	53.49 (↓ 0.16)	91.35 (↓ 0.15)	84.28 (↓ 0.16)	94.30 (↓ 0.19)	91.11 (↓ 0.31)	84.35 (↓ 0.21)
WAC without CT and SC	91.30 (↓ 0.56)	52.88 (↓ 0.77)	91.16 (↓ 0.34)	84.05 (↓ 0.39)	94.17 (↓ 0.32)	90.49 (↓ 0.93)	84.00 (↓ 0.56)

CE stands for cross entropy. Performances degradation compared to the original WAC is indicated by ↓ symbol.

TABLE XII
Human Evaluation on the Augmented Sentences

Method	Fluency	Semantic Similarity
DWB	2.31	3.09
PWWS	**2.9**	2.77
TextBugger	2.65	**3.14**
WAC	2.05	2.44

series of word-based augmentation schemes become challenging for readers to understand.

Even though the semantic similarity in a subjective evaluation is low, the main idea of the proposed WAC focuses on the semantic similarity in latent space as shown by (11). The generated sentences have been demonstrated to enhance the capacity of the trained classifier to handle unseen noisy sentences, as evidenced by the results in Tables VI and VII. Such a phenomenon potentially aligns with the worst-case data augmentation scenario. High fluency or semantic similarity of the augmented data does not necessarily guarantee improvement in the model robustness against unseen noises.

E. Computational Complexity Evaluation

The computational complexity was evaluated by comparing different works with focus only on contextual augmentation methods for fair comparison. Firstly, compared to the previous works applying the NAS method [14], WAC has much lower computational cost. This can be attributed to the WAC setting, which enables the agent to receive the reward immediately after the action is executed. Consequently, unlike the NAS learning setting, there is no necessity for WAC to await the convergence of the child network to obtain the reward. Next, WAC is compared with the greedy methods in word-level augmentation [6], [11], [13]. Let's define $\mathbf{x}^\star$ as the optimum perturbed sentence. To obtain $\mathbf{x}^\star$, the greedy search method should find a word in vocabulary set $\mathcal{V}$ with size $|\mathcal{V}|$ that can optimally perturb the target model. This implies that the greedy method must execute at most $O(|\mathcal{V}|)$ queries in each step. Assume that there are N_w words in an input sentence and L steps are required to get $\mathbf{x}^\star$. The computational complexity considering L steps becomes $O((|\mathcal{V}| \cdot N_w)^L)$. Then, it is evident that the computational complexity of the greedy solution grows exponentially with the number of steps L. Furthermore, it is important to note that the aforementioned computational complexity is grounded only on a single augmentation strategy, specifically the *word substitution*. The actual complexity would be higher when the greedy method selects different augmentation strategies at each step and applies them to the target word. This substantial computational demand is also reflected in the previous research employing the greedy search for character-level augmentation [7], [8].

On the other hand, the complexity of the WAC is substantially reduced because WAC does not need to try all of the possible scenarios, like greedy search, to produce the worst-case augmentation. Instead, data augmentation is directly generated by the model after learning from the reward received to get the best strategy for data generation. In the evaluation, WAC demonstrated a reduced computation time in practical implementation. For instance, in the SST-2 dataset, WAC required only 234 minutes to generate distinctive adversarial data with multiple transformation strategies. In comparison, methods like DWB [8], PWWS [6], and TextBugger [7], all of which implemented a single transformation strategy via greedy search, required 257 minutes, 625 minutes, and 283 minutes, respectively. These computation were measured by using a personal computer equipped with a NVIDIA RTX 2080TI GPU, a 19-10900 K CPU, and 128 GB Memory.

F. Ablation Study

To evaluate the importance of individual objectives in the proposed WAC, the ablation study was conducted by individually evaluating each objective in the strategy search for worst-case augmentation. We set the reward setting by using standard cross-entropy instead of information bottleneck approach. The results are shown by Table XI (top), where the performance degradation is consistently observed in all test sets. This indicates that the IB based reward was helpful for generating the label preserving data in worst-case augmentation. Simultaneously, an ablation study concerning the defense strategy is depicted in Table XI (bottom). The results show that CT notably contributed to the robustness of the model indicated by significant performance drop if CT loss is taken out from the optimization. Meanwhile, SC loss just gave slight improvement in model robustness. However, the combination of CT and SC losses considerably enhances the model robustness, especially in the SST-5, CR and TREC-6 datasets. The source codes corresponding to this study can be accessed via https://github.com/NYCU-MLLab/.

VI. Conclusion

This paper has presented a new adversarial worst-case augmentation to improve the robustness of model classification in presence of various noises. Different from the previous adversarial data augmentation, both attack and defense strategies were designed carefully in order to achieve not only the meaningful data augmentation but also the robust classification by utilizing the augmented data properly. The worst-case augmenter and classifier were jointly trained to fulfill attack and defense with the perspectives of exploration and exploitation from reinforcement learning, respectively. The attack method was built based on a reinforcement learning algorithm to enable both distinctive augmentation strategies and low computation training cost. For the defense strategy, instead of using the generated examples only as the additional training dataset and doing a standard classification learning, a new approach based on information bottleneck objective with additional local and global regularization were implemented for uncertainty modeling. An adversarial and informative RL solution to efficient augmentation and robust classification was constructed. From a series of experimental results, the proposed worst-case augmentation and classification showed better robustness in six different text classification tasks over ten different perturbations compared to the strong adversarial baseline methods.

References

[1] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[2] H. Lio, S.-E. Li, and J.-T. Chien, "Adversarial mask transformer for sequential learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 4178–4182.

[3] C.-E. Hsu, M. Rohmatillah, and J.-T. Chien, "Multitask generative adversarial imitation learning for multi-domain dialogue system," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 954–961.

[4] J.-T. Chien and Y.-H. Huang, "Latent semantic and disentangled attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10047–10059, Dec. 2024.

[5] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2890–2896.

[6] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1085–1097.

[7] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[8] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *Proc. IEEE Secur. Privacy Workshops*, 2018, pp. 50–56.

[9] J.-T. Chien and Y.-A. Chen, "Self-supervised adversarial training for contrastive sentence embedding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[10] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4324–4333.

[11] H. Liu et al., "SSPAttack: A simple and sweet paradigm for black-box hard-label textual adversarial attack," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 11, pp. 13228–13235.

[12] Z. Shao, Z. Wu, and M. Huang, "AdvExpander: Generating natural language adversarial examples by expanding text," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1184–1196, 2022.

[13] S. Liu, N. Lu, C. Chen, and K. Tang, "Efficient combinatorial optimization for word-level adversarial textual attack," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 98–111, 2022.

[14] T. Niu and M. Bansal, "Automatically learning data augmentation policies for dialogue tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 1317–1323.

[15] J. Xu, L. Zhao, H. Yan, Q. Zeng, Y. Liang, and X. Sun, "LexicalAT: Lexical-based adversarial reinforcement training for robust sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 5518–5527.

[16] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 113–123.

[17] M. Rohmatillah and J.-T. Chien, "Hierarchical reinforcement learning with guidance for multi-domain dialogue policy," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 748–761, 2023.

[18] Y. Qu, D. Shen, Y. Shen, S. Sajeev, W. Chen, and J. Han, "CoDA: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding," in *Proc. Int. Conf. Learn. Representations*, 2021.

[19] C.-T. Chu, M. Rohmatillah, C. Lee, and J.-T. Chien, "Augmentation strategy optimization for language understanding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2022, pp. 7952–7956.

[20] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 229–256, 1992.

[21] L. Zhao, T. Liu, X. Peng, and D. Metaxas, "Maximum-entropy adversarial data augmentation for improved generalization and robustness," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14435–14447.

[22] J.-T. Chien and S.-T. Chang, "Bayesian asymmetric quantized neural networks," *Pattern Recognit.*, vol. 139, 2023, Art. no. 109463.

[23] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12556–12565.

[24] F. Qiao and X. Peng, "Uncertainty-guided model generalization to unseen domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6790–6800.

[25] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 658–666.

[26] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, "Adversarial autoaugment," in *Proc. Int. Conf. Learn. Representations*, 2020.

[27] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 86–96.

[28] Z. Xie et al., "Data noising as smoothing in neural network language models," in *Proc. Int. Conf. Learn. Representations*, 2017.

[29] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 6382–6388.

[30] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "DeCLUTR: Deep contrastive learning for unsupervised textual representations," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 879–895.

[31] J.-T. Chien and W.-Y. Sun, "Adversarial augmentation for adapter learning," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2023, pp. 1–7.

[32] A. Gupta et al., "Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 13981–13998.

[33] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[34] J. Chen, R. Zhang, Z. Luo, C. Hu, and Y. Mao, "Adversarial word dilution as text data augmentation in low-resource regime," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 11, pp. 12626–12634.

[35] P. Zhan, J. Yang, H. Wang, C. Zheng, X. Huang, and L. Wang, "Similarizing the influence of words with contrastive learning to defend word-level adversarial text attack," in *Proc. Conf. Findings Assoc. Comput. Linguistics*, 2023, pp. 7891–7906.

[36] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[37] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6256–6268.

[38] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[39] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5339–5349.

[40] J. Blanchet and K. Murthy, “Quantifying distributional model risk via optimal transport,” *Math. Operations Res.*, vol. 44, no. 2, pp. 565–600, 2019.

[41] S. Boyd, S. P. Boyd, and L. Van den Berghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[42] O. Nachum, S. S. Gu, H. Lee, and S. Levine, “Data-efficient hierarchical reinforcement learning,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 3303–3313.

[43] Y. Tu, M.-W. Mak, and J.-T. Chien, “Variational domain adversarial learning with mutual information maximization for speaker verification,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2013–2024, 2020.

[44] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley-Interscience, 2006.

[45] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[46] J.-T. Chien and Y.-A. Chen, “Towards a unified view of adversarial training: A contrastive perspective,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2024, pp. 5365–5369.

[47] J.-T. Chien and C.-W. Huang, “Stochastic adversarial learning for domain adaptation,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–7.

[48] L. Li, M.-W. Mak, and J.-T. Chien, “Contrastive adversarial domain adaptation networks for speaker recognition,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 2236–2245, May 2022.

[49] J.-T. Chien and K.-T. Peng, “Neural adversarial learning for speaker recognition,” *Comput. Speech Lang.*, vol. 58, pp. 422–440, 2019.

[50] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, “Improving the robustness of deep neural networks via stability training,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4480–4488.

[51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015.

[52] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. Int. Conf. Learn. Representations*, 2019.

[53] R. Socher et al., “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[54] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 168–177.

[55] T. A. Wilson, *Fine-Grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States*. Pittsburgh, PA, USA: Univ. of Pittsburgh Press, 2008.

[56] X. Li and D. Roth, “Learning question classifiers,” in *Proc. Int. Conf. Comput. Linguistics*, 2002, pp. 1–7.

[57] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput.*, J. Burstein, C. Doran, and T. Solorio, Eds., Jun. 2019, pp. 4171–4186.

[59] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[60] D. Cer et al., “Universal sentence encoder for english,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 169–174.

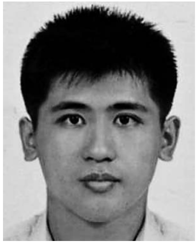
[61] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, “Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 119–126.

[62] D. Li et al., “Contextualized perturbation for textual adversarial attack,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2021, pp. 5053–5069.

[63] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond accuracy: Behavioral testing of NLP models with checklist,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4902–4912.

Jen-Tzung Chien (Senior Member, IEEE) is currently the Lifetime Chair Professor in National Yang Ming Chiao Tung University, Hsinchu, Taiwan. He has authored more than 250 peer-reviewed articles in machine learning, deep learning, and Bayesian learning with applications on natural language processing and computer vision, and three books including *Bayesian Speech and Language Processing*, Cambridge University Press, 2015, *Source Separation and Machine Learning*, Academic Press, 2018, and *Machine Learning for Speaker Recognition*, Cambridge University Press, 2020. He was a Tutorial Speaker of AAAI, IJCAI, ACL, MM, KDD, ICASSP, CIKM, WSDM, COLING and Interspeech. He was the recipient of the Best Paper Award in IEEE Workshop on Automatic Speech Recognition and Understanding in 2011, and IEEE International Workshop on Machine Learning for Signal Processing in 2023.

Mahdin Rohmatillah received the Ph.D. degree from National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2024. He is an Assistant Professor in Universitas Brawijaya, Malang, Indonesia. His research interests include machine learning, reinforcement learning, and dialogue system.



Chang-Ting Chu received the B.S. and M.S. degrees in electrical and computer engineering from National Chengchi University, Taipei, Taiwan, in 2019 and 2021, respectively. His research interests include adversarial learning and natural language processing.