AUGMENTATION STRATEGY OPTIMIZATION FOR LANGUAGE UNDERSTANDING

Chang-Ting Chu[⋆] *Mahdin Rohmatillah*[⋆] *Ching-hsien Lee*[†] *Jen-Tzung Chien*[⋆]

[⋆]Dept of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Taiwan
[†]Computational Intelligence Technology Center, Industrial Technology Research Institute, Taiwan

ABSTRACT

This paper presents a new language processing and understanding where an adaptive data augmentation strategy for individual documents is proposed instead of using one universal policy for the whole dataset. Importantly, a reinforcement learning and understanding method is exploited for document classification where the document encoder, augmenter and classifier are jointly optimized. In particular, a new reward function based on the consistency loss maximization is presented to assure the diversity of the generated documents. Using this method, the reward for adaptive augmentation policy is immediately calculated for every augmented instance without the need of waiting the child model performance metrics as the reward. The experiments on various classification tasks with a strong baseline model show that the augmentation strategy optimization can improve the model training process by providing meaningful augmentation data which eventually result in desirable evaluation performance. Furthermore, the extensive studies on the behavior of policy in different settings are provided in order to assure the diversity of the augmented data that was obtained by the proposed method.

***Index Terms*—** Data augmentation, policy optimization, document representation, natural language understanding

1. INTRODUCTION

Due to the success of automated machine learning (AutoML), a lot of efforts have been conducted to build an automated search process for augmentation strategy. In this research direction, the first popular method called AutoAugment [1] was proposed to automatically estimate the augmentation policy given by a target dataset and specified by a model type by using the REINFORCE algorithm [2]. Unfortunately, AutoAugment required thousands of graphical processing unit (GPU) hours in computer vision (CV) tasks due to its evaluation strategy, where the validation accuracy was obtained as the reward once child model training was done by using the policy sampled from a controller based on recurrent neural network. Inspired by AutoAugment, this paper proposes the automated data augmentation for language processing and understanding [3, 4, 5] with twofold novelties. First, an adaptive augmentation policy is optimized for individual datapoints, sentences or documents. Second, an efficient evaluation strategy, designed for training the auto augmentation policy network, is developed. In order to achieve the aforementioned novelties, an augmenter or agent is trained via a reinforcement learning (RL) problem which is formulated by using the states for documents, the actions for augmentation and the rewards for policy evaluation [6, 7]. Accordingly, a state encoder is introduced to find document embedding. In order to collect a pool of semantic-preserving text augmentations, the operation settings proposed by easy data augmentation (EDA) [8], consisting of four simple but powerful operations including synonym replacement, random insertion, random swap, and random deletion are adopted as an action setting for the agent. The augmentation policy network is proposed to learn how to arrange a sequence of operations to synthesize the documents for improving the generalization of text representation. Different from EDA [8] using a single augmentation for a sentence, an automatic stack of distinct actions [9] is proposed to provide meaningful additional data. In addition, the reward setting is developed by enhancing the generalization through hard positive examples [10] based on the consistency loss maximization [9]. This setting can improve the semantic-preserving of augmented documents based on cross entropy loss maximization in [10] especially for short documents. Based on the experimental results, the proposed method successfully integrates with the strong baseline model using RoBERTa [11] and obtains a notable improvement for language understanding in different document classification tasks. Furthermore, an investigation for the augmented data and the behavior of the policy is addressed.

2. BACKGROUND SURVEY

In recent years, the issue of data augmentation [12] has been attracting high interest since state-of-the-art (SOTA) results were achieved, especially in CV domain. AutoAugment [1] was constructed by combining several automatic augmentation procedures, and successfully achieved SOTA results in image classification tasks including CIFAR-10, CIFAR-100, SVHN, and ImageNet. In the further investigations, AutoAugment based methods also brought substantial benefits in self-supervised learning in which data augmentation played a crucial role in defining an effective prediction task. The method called SimCLRv2 [13] was the first self-supervised learning through augmentation policy of images. More recently, many researchers attempted to extend data augmentation methods to natural language tasks by utilizing the previously-found tricks to text augmentation, e.g. back-translation [14], noising augmentation [15], and EDA. However, most of the recent results, as given in [16], only applied the single policy like back-translation and the distance calculation of anchor document relative to the whole dataset in a self-supervised learning task. The resulting system caused a limited performance. The only effort that implemented the data augmentation using AutoAugment was found in [17] which was applied and combined in a dialogue task. However, this method was very time-consuming since it required to wait the child network to be converged to receive the reward for policy optimization. Different from the previous works, a new approach is proposed to find an adaptive data augmentation for each individual document without requiring the learning convergence in child network. The effectiveness of this approach is evaluated by showing the generalization capability and the classification performance for input documents. An algorithm for augmentation strategy optimization is implemented for the whole system.

3. AUGMENTATION STRATEGY OPTIMIZATION

Figure 1 shows an overview of the proposed method consisting of document encoder, document classifier, and REINFORCE augmenter which are formed with the parameters θ_e, θ_c, and θ_a, respectively. Document encoder is used to find a document embedding from an input text document which is fed to text classifier to make an appropriate prediction, optimized by minimizing the cross entropy loss. Inspired by the hard positive example generated for self-supervised learning in [10], an REINFORCE augmenter is used to improve model generalization for document representation by generating hard positive examples which are optimized by using a reward function based on the consistency loss of the prediction between augmented document and input document.

3.1. Augmenter Learning

Considering the aspect of RL, the policy gradient method based on REINFORCE [2] is introduced to train an augmentation policy network where the state, action, and reward function are carefully designed and formulated.

3.1.1. State

In this work, the state s_0 represents the document embedding obtained by $s_0 = f(x_0^{(i)}; \theta_e)$, where $x_0^{(i)}$ is the i^{th} input document. s_t is the state of augmented document after being augmented by t times either using identical or different actions.

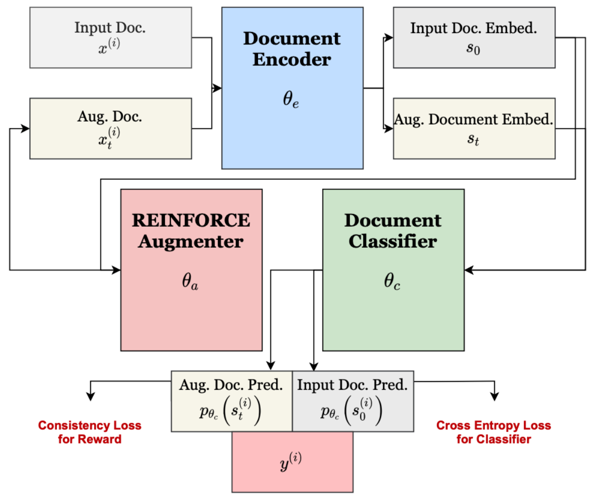

Fig. 1: Overview of the proposed method. REINFORCE augmenter is assigned for training an augmentation policy network to generate the augmented document.

3.1.2. Action

There are five discrete actions in the action set given as $A = \{a_t\} = \{\text{SR}, \text{SI}, \text{RD}, \text{RS}, \text{STOP}\}$, where each action abbreviation meaning can be shown in Table 2. Action STOP means that agent decides to stop augmenting the document. The illustration of each action is shown by Table 1. By assigning individual action in each time step, a series of stacked augmentations can be formed to help model generalization.

Table 1: Examples of augmented and original sentences via four augmentations and one stop given by the action labels.

Label: Action	Sentence
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 insertion)	Sparse only curiously **oddly** compelling
4: **Stop**	Sparse only curiously compelling

3.1.3. Reward

Reward for evaluating the augmentation policy network is defined by measuring the consistency loss between the prediction of original and augmented examples. To pursue the generalization of augmented document, the reward for strategy optimization is measured by the loss based on the Jensen-Shannon (JS) divergence [9, 18] which is bounded and more stable relative to Kullback-Leibler (KL) divergence

$$\text{JS}(p_{\theta_c}(s_t), p_{\theta_c}(s_0)) = \frac{1}{2}(\text{KL}(p_{\theta_c}(s_t)\|\mathcal{M}) + \text{KL}(p_{\theta_c}(s_0)\|\mathcal{M})) \tag{1}$$

where $\mathcal{M} = \frac{1}{2}(p_{\theta_c}(s_t) + p_{\theta_c}(s_0))$ with the prediction likelihood $p_{\theta_c}(s_t)$ using parameter θ_c. To obtain meaningful transformations, a penalty reward ε is used to suppress the transformation leading to semantic changes indicated by cosine similarity between s_0 and s_t which is lower than a threshold α. ε equals to zero is selected following common setup in RL tasks. Reward at each time step t is then defined as

$$r_t = \begin{cases} \varepsilon, & \text{if } \cos(s_t, s_0) < \alpha \\ \mathrm{JS}(p_{\theta_c}(s_t), p_{\theta_c}(s_0)), & \text{else.} \end{cases} \quad (2)$$

Here, p_{θ_c} is the classification network measuring the label prediction for a document state s_t. Following (2), the action a_t for a document s_t semantically close to original document s_0 with the diverse prediction is chosen. In this scenario, policy is optimized to refine the misclassified input through a classifier learning. However, the experiments revealed that this reward setting encouraged the policy to only conduct safe actions. To mitigate this issue, an additional penalty reward ρ_t is introduced to further increase the prediction diversity

$$r_t \leftarrow r_t - \rho_t, \quad \text{where } \rho_t = \frac{\bar{r}t}{T} \quad (3)$$

where ρ_t is formed as the average reward $\bar{r}$ times the current step t divided by the maximum number of steps T which is specified as nine in this work. A new reward is fulfilled.

3.1.4. Learning Algorithm

In learning procedure, it is essential to freeze the parameters of document encoder θ_e and classifier θ_c during the learning of policy $\pi_{\theta_a}(a_t|s_t)$ with parameter θ_a. The training of REINFORCE augmenter is shown by Algorithm 1 where γ is a decay factor. The state evolved from s_{t-1} to s_t is addressed as follows. s_{t-1} is the embedding of the augmented document $x^{(i)}_{t-1}$ with $t-1$ times. π_{θ_a} produces the action a_{t-1} given s_{t-1} which is then applied to $x^{(i)}_{t-1}$ to transform it into a new augmented document $x^{(i)}_t$. After all, we obtain the document s_t by encoder $f(x^{(i)}_t; \theta_e)$. This process is iteratively performed until termination for consistency loss maximization.

3.2. Classifier Learning

Importantly, the classifier is jointly trained with augmenter so as to search the transformation for augmented data $\{\widehat{x}^{(i)}\}$ from original data $\{x^{(i)}\}$ by preserving the labels $\{y^{(i)}\}$ and the semantics. Misclassified document can be refined. To meet such an objective, the parameters of encoder and classifier $\theta = \{\theta_e, \theta_c\}$ are jointly optimized by minimizing the cross entropy (CE) loss as well as the consistency loss via

$$\theta^* = \arg\min_\theta \sum_{x^{(i)}, y^{(i)} \in \mathcal{D}} \mathcal{L}_{\text{CE}}(p_\theta(x^{(i)}), y^{(i)}) + \mathcal{L}_{\text{CE}}(p_\theta(\widehat{x}^{(i)}), y^{(i)}) + \mathcal{L}_{\text{consist}}(p_\theta(x^{(i)}), p_\theta(\widehat{x}^{(i)})). \quad (4)$$

Similar to generative adversarial network [19, 20, 21], a kind of adversarial learning is performed to minimize $\mathcal{L}_{\text{consist}}(\cdot)$ for discriminator and maximize it for REINFORCE augmenter. Augmentation strategy is implemented in a way of finding the *most diverse augmentation* with the *least classification inconsistency* between augmented and original documents.

Algorithm 1: Training for augmentation strategy

Require: $\mathcal{D}$ training dataset. η learning rate
T maximum number of steps
θ_e, θ_a, pars of encoder and augmenter

while θ_a *is not converged* **do**
  for *i=1, ..., $|\mathcal{D}|$* **do**
    input $x^{(i)}_0$ as i^{th} document in $\mathcal{D}$
    $s_0 = f(x^{(i)}_0; \theta_e)$ as the embedding of $x^{(i)}_0$
    $\{s_0, a_0, r_0, \ldots s_{T-1}, a_{T-1}, r_{T-1}\} \sim \pi_{\theta_a}(\tau)$
    $G_t = \sum_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'}$
    $g_{\theta_a} \leftarrow \nabla_{\theta_a} \sum_{t=0}^{T-1} \log \pi_{\theta_a}(a_t|s_t) \cdot G_t$
    $\theta_a \leftarrow \theta_a + \eta \cdot \mathrm{Adam}(\theta_a, g_{\theta_a})$

4. EXPERIMENTS

The evaluation was conducted in several text classification tasks including Stanford sentiment treebank (SST) for both binary setting and fine-grained setting [25], customer review (CR) [26], multi-perspective question answering (MPQA) [27], subjectivity (Subj) [28], and TREC-6 [29]. Investigation of augmentation policy behavior and evaluation of system performance are described in what follows.

4.1. Evaluation on Policy Optimization

First of all, Table 2 shows the augmented data and the conducted actions for the selected documents in SST. It is intriguing to see that the policy performs much better on the short documents pointed out by their readabilities and JS-based rewards. There are two major factors which trigger this phenomenon. The first one is due to the document distribution in SST which is dominated by short documents, causing the augmentation policy could learn well over the short documents but failed to learn over the long documents. Another factor is the natural property of long documents which are resistant to few augmentation actions, forcing agent to explore more actions in order to obtain diverse augmentations. Different from previous method using single augmentation [8], the proposed method conducts the stacked data augmentation (SDA) for individual documents. Another important finding is the effect introduced by different similarity threshold values α, shown by Table 3 (left). Due to different threshold settings, the distributions of four augmentation actions are different.

Table 2: Illustration for the proposed stacked data augmentation (SDA) with five actions (0: random delete, 1: random swap, 2: random synonym replacement, 3: random synonym insertion, 4: stop operation). "x" denotes the failed action due to losing of original semantic meaning, indicated by the condition $\cos(s_t, s_0) < \alpha$. The order of actions and the received reward are shown.

Original Document	**Augmented Document**	**Action**	**Reward**
The name says it all.	The name pronounce it totally	2 2 0 4	0.0484
A lovely and beautifully photographed romance.	A take shoot photograph and lovely beautifully	3 1 2 0 3 4	0.0091
Rouge is less about a superficial midlife crisis than it is about the need to stay in touch with your own skin, at 18 or 80.	with skin. it than less about at or your the ain superficial is midlife crisis, stay need sense of touch touch contain is in about to Rouge vitamin a 18	1 2 2 3 2 3 0 2 3 1 x	0.0049

Table 3: (left) Distribution of actions taken by the policy. **Sim.Thr.** stands for similarity threshold. **Stop** indicates the successfully augmented document without exceeding the max step T or violating the similarity threshold α. (right) Accuracy (%) on different classification tasks. The results from reference papers are shown. "-" denotes the missing results. Augmentation methods using EDA [8], and back-translation (Back) [14] are compared with SDA. pre-trained model using RoBERTa is merged.

Sim.Thr.	0.7	0.8	0.9
Delete	8.3%	3.8%	3.7%
Swap	0.8%	4.9%	8.0%
Replace	39.8%	22.4%	**67.2%**
Insert	**51.1%**	**68.6%**	20.8%
Stop	7.5%	20.8%	29.1%

Model	**SST-2**	**SST-5**	**CR**	**MPQA**	**Subj**	**TREC**
EFL [22]	**96.9**	-	92.5	90.8	97.1	-
byte mLSTM [23]	91.7	54.6	90.6	88.8	94.7	90.4
BERT [24]	93.1	55.5	-	-	**97.3**	96.8
RoBERTa [11]	94.8	56.6	93.2	90.4	96.0	96.8
RoBERTa with EDA	94.6	56.9	93.3	90.0	95.3	96.6
RoBERTa with Back	95.0	57.3	94.1	90.9	96.9	**97.4**
RoBERTa with SDA	95.2	**58.6**	**94.7**	**91.4**	96.0	97.0

In particular, the action of insertion is found to be dominant in the settings of 0.7 and 0.8 while the action of synonym replacement is dominant in 0.9 as random synonym replacement is the safest action among four actions. It is favored to use higher α for more balanced actions. Meanwhile, an interesting discovery is found during the experiment using the setting of 0.7 whereas the agent tends to perform high risk action such as random deletion. Furthermore, different values α also affect the probability of successfully stop action as high threshold makes the agent to avoid risky actions. Therefore, the probability of the successfully stop action becomes higher when using higher α. The setting of 0.9 is used hereafter.

4.2. Evaluation on Text Classification

Instead of using simple testbed to to validate the effectiveness of the proposed augmentation strategies, in this work, RoBERTa [11] was used as the strong pre-trained model. The encoder in SDA was fine-tuned by using RoBERTa. Table 3 (right) shows that SDA significantly outperforms the strong pre-trained models using byte mLSTM [23], BERT [24] and RoBERTa in most tasks. In CR task, the SDA even performs better that the larger pre-trained model like EFL [22], built based on RoBERTa large [11]. For most tasks, the SDA obtained higher accuracy than two other augmentation methods using EDA [8] and back-translation [14, 30] where the RoBERTa encoder is commonly applied. By using the strong testbed, the benefit of using SDA is obvious. In the comparison, the proposed method just slightly outperformed in two datasets, subjectivity and TREC-6, due to the abundance of the datasets which means additional data is not that necessary to help model to achieve convincing performance. For the ablation study, in order to realize whether the performance improvement is introduced from stacking different augmentation with consistency loss maximization or not, the proposed method is compared to stacking augmentation methods randomly. We obtained 94.7% and 57.1% in SST-2 and SST-5, respectively, for stacking randomly. The ablation study on stacking reveals the effectiveness of the proposed method. The implementation code and hyperparameters are available on https://github.com/NYCU-MLLab/Augmentation-Strategy-Optimization-for-Language-Understanding.

5. CONCLUSIONS

A new method for searching the stacked distinct augmentation actions has been presented to employ in six text classification tasks. The results showed that the generalization of the model with strong language understanding module could be further improved with the proposed method. Moreover, the augmentation policy could generate some readable sentences and behaved diversely in different settings of REINFORCE augmenter. However, further investigation is required in order to tackle the sensitivity of hyperparameter setting.

6. REFERENCES

[1] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Pro. of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

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

[3] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[4] J.-T. Chien, "Deep Bayesian mining, learning and understanding," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 3197–3198.

[5] J.-T. Chien, "Deep Bayesian natural language processing," in *Proc. of Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2019, pp. 25–30.

[6] M. Rohmatillah and J.-T. Chien, "Causal confusion reduction for robust multi-domain dialogue policy," in *Proc. of Annual Conference of the International Speech Communication Association*, 2021, pp. 3221–3225.

[7] M. Rohmatillah and J.-T. Chien, "Corrective guidance and learning for dialogue management," in *Proc. of ACM International Conference on Information & Knowledge Management*, 2021, pp. 1548–1557.

[8] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 6382–6388.

[9] 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. of International Conference on Learning Representations*, 2021.

[10] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2606–2615.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[12] J.-T. Chien and K.-T. Peng, "Adversarial learning and augmentation for speaker recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2018, pp. 342–348.

[13] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22243–22255, 2020.

[14] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 86–96.

[15] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng, "Data noising as smoothing in neural network language models," *arXiv preprint arXiv:1703.02573*, 2017.

[16] H. Fang and P. Xie, "CERT: Contrastive self-supervised learning for language understanding," *arXiv preprint arXiv:2005.12766*, 2020.

[17] T. Niu and M. Bansal, "Automatically learning data augmentation policies for dialogue tasks," *arXiv preprint arXiv:1909.12868*, 2019.

[18] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4480–4488.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

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

[21] Y. Tu, M.-W. Mak, and J.-T. Chien, "Variational domain adversarial learning with mutual information maximization for speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2013–2024, 2020.

[22] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, "Entailment as few-shot learner," *arXiv preprint arXiv:2104.14690*, 2021.

[23] M. Khodak, N. Saunshi, Y. Liang, T. Ma, B. Stewart, and S. Arora, "A La Carte embedding: Cheap but effective induction of semantic feature vectors," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 12–22.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.

[26] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. of ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.

[27] T. A. Wilson, *Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states*, University of Pittsburgh, 2008.

[28] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. of Annual Meeting on Association for Computational Linguistics*, 2004, pp. 271–278.

[29] X. Li and D. Roth, "Learning question classifiers," in *Proc. of International Conference on Computational Linguistics*, 2002.

[30] H. Lio, S.-E. Li, and J.-T. Chien, "Adversarial mask transformer for sequential learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.