Class-Incremental Learning based on Label Generation

Yijia Shao[1], **Yiduo Guo**[1], **Dongyan Zhao**[1,2,3] **and Bing Liu**[4]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]National Key Laboratory of General Artificial Intelligence [3]BIGAI, Beijing, China
[4]Department of Computer Science, University of Illinois at Chicago
shaoyj@pku.edu.cn, yiduo@stu.pku.edu.cn, zhaody@pku.edu.cn, liub@uic.edu

Abstract

Despite the great success of pre-trained language models, it is still a challenge to use these models for continual learning, especially for the *class-incremental learning* (CIL) setting due to *catastrophic forgetting* (CF). This paper reports our finding that if we formulate CIL as a *continual label generation* problem, CF is drastically reduced and the generalizable representations of pre-trained models can be better retained. We thus propose a new CIL method (VAG) that also leverages the sparsity of vocabulary to focus the generation and creates pseudo-replay samples by using label semantics. Experimental results show that VAG outperforms baselines by a large margin.[1]

1 Introduction

Large pre-trained language models (PLMs) have become the *de facto* standard in building NLP systems. However, how to best use them for continual learning (CL) is still a significant question (Huang et al., 2021; Xia et al., 2022; Pasunuru et al., 2021; Ke et al., 2021). Many existing studies focus on *task-incremental learning* (TIL) where the model learns distinct tasks sequentially and is given the task identity for inference. These works usually keep the PLM unchanged and update a series of additional structures such as adapters (Gururangan et al., 2022) or prompts (Zhu et al., 2022; Qin and Joty, 2022). Though effective, these methods cannot be used in a more challenging setting of **class-incremental learning** (CIL) which does not provide task information at test time.

CIL aims to build a single model to make predictions over incrementally learned classes organized as tasks (formal definition in §2). Wu et al. (2022) conducted a comparative study on PLM in CL and showed that PLMs perform extremely poorly in the CIL setting due to *catastrophic forgetting* (CF)[2]. Also, as the task information is unknown, CIL further requires the model to predict the task identity of each test instance correctly.

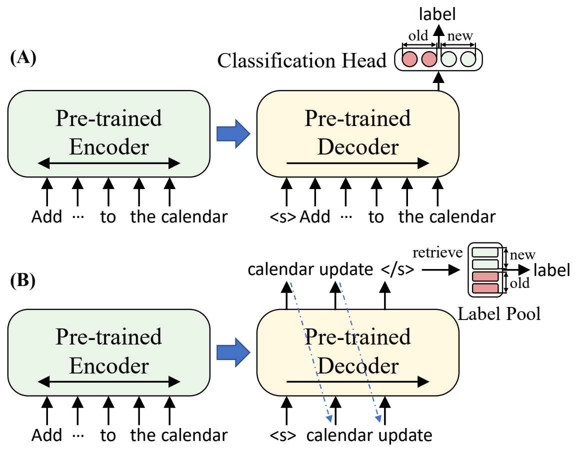

Figure 1: Comparison between classifier framework (A) and generation framework (B) of using a pre-trained encoder-decoder model for class-incremental learning.

In this work, we re-examine the problem of using PLM for CIL and discovered that *formulating CIL as* ***continual label generation*** *can greatly improve PLMs' continual learning ability*. As illustrated in Figure 1, a traditional classifier views the PLM as a large feature extractor and uses a linear classification head to map the extracted features to a probability distribution on both old and new labels. However, we can also use a generation approach to directly fine-tune the PLM to generate a label sequence (indicating a label) for a test instance. The final label is retrieved from the label pool of the classes learned so far based on text similarity.

Some existing CL works have leveraged generation. For example, LAMOL (Sun et al., 2019) is a TIL system that uses generation to unify different types of tasks and creates pseudo replay samples;

[1] Our code is publicly available at https://github.com/shaoyijia/VAG.

[2] CF means that a neural network forgets previously learned knowledge when trained on new tasks, resulting in a decline in performance on earlier tasks (McCloskey and Cohen, 1989).

Zhang et al. (2022) focuses on the continual learning of different generation tasks.[3] Different from these works, we are the first to directly use the generation objective to effectively ease the CF issue in the CIL process. Our experiments demonstrate that the generation objective is more suitable to the continual learning of PLM. To study the inner working of the paradigm shift, in §3.1, we quantitatively show that the generation objective can prevent the PLM from representation collapse (Aghajanyan et al., 2021), thus preserving its ability to continually learn new classes.

To further improve the generation approach, we propose the **VAG** (**V**ocabulary-**A**ware Label **G**eneration) system for CIL. VAG modifies the generation loss by focusing on different vocabulary subsets when learning different tasks. Owning to the natural sparsity of vocabulary, the modified loss leads to a sparse model update that greatly eases the CF issue. Moreover, VAG exploits the label semantics to create pseudo replay data via a label-based augmentation. Extensive experiments on 5 datasets show that VAG drastically outperforms baselines in non-exemplar based CIL (*i.e.*, without saving any replay sample) and also achieves better results when a small amount of saved replay data is used.

2 Background

Class-Incremental Learning (CIL). CIL learns a sequence of tasks $\{1, ..., T\}$ incrementally (Kim et al., 2022). Each task t learns a set of new classes $\mathcal{C}_t$. At task $t \in \{1, ..., T\}$, the system is given a training set $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$, where $\mathcal{X}_t = \{x_j^{(t)}\}_{j=1}^{N_t}$ is the input data, $\mathcal{Y}_t = \{y_j^{(t)}\}_{j=1}^{N_t}$ is the set of their class labels and $y_j^{(t)} \in \mathcal{C}_t$. The classes in different tasks are disjoint, $\mathcal{C}_t \cap \mathcal{C}_{t'} = \emptyset, \forall t' \neq t$. At inference, given a test instance, the system selects a class label from $\bigcup_{t=1}^{T} \mathcal{C}_t$ *without knowing the task identity*. The performance of the system is evaluated in the accuracy of the test samples from all seen classes.

Encoder-Decoder Model Encoder-decoder models take a sequence of tokens as input $X = x_1, ..., x_n$ and generate the target sequence $Y = y_1, ..., y_m$ in an auto-regressive manner. Specifically, the encoder maps the input sequence to a vector representation $c = f_{\theta_{enc}}(X) \in \mathbb{R}^{d_{enc}}$. Suppose the auto-regressive decoder has already generated $Y_{1:i-1} = y_1, ..., y_{i-1}$, the next-token probability is

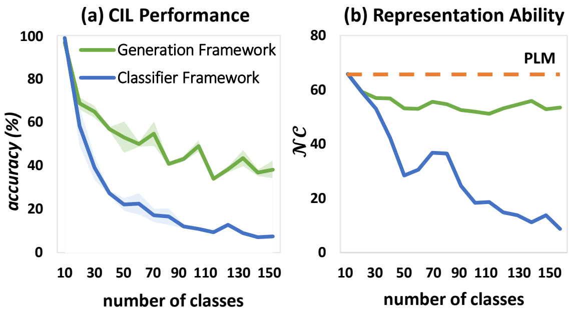

Figure 2: Accuracy (%) and $\mathcal{NC}$ (neural collapse) comparison of the classifier framework and generation framework for CIL on CLINC150 (15 tasks). For both *accuracy* and $\mathcal{NC}$, higher numbers are better.

$$P(y_i|c, Y_{1:i-1}) = \frac{\exp(E_{y_i}^{\mathsf{T}} f_{\theta_{dec}}(c, Y_{1:i-1}))}{\sum_{w \in \mathcal{V}} \exp(E_w^{\mathsf{T}} f_{\theta_{dec}}(c, Y_{1:i-1}))}. \quad (1)$$

Here, $E_w \in \mathbb{R}^{d_{dec}}$ denotes the word embedding of token $w \in \mathcal{V}$, where $\mathcal{V}$ is the model vocabulary. The model parameters are optimized to minimize the negative log-likelihood of ground truth y_t.

3 VAG System

We present the proposed VAG system which reframes CIL as a continual label generation problem. Figure 3 gives an overview of VAG with two major components.

3.1 Classification via Generation

VAG solves classification via label generation and maintains a label pool $\mathcal{P}$ of label sequences. Each label $c \in \mathcal{C}_t$ is a sequence of tokens representing a class label. When training task t, instead of mapping $\mathcal{C}_t$ to integer indexes representing class labels, VAG retains the label semantics and fine-tunes the PLM $\mathcal{M}$ to generate the label sequence conditioned on the input sequence $x_j^{(t)}$. In the CIL process, $\mathcal{P}$ keeps growing to contain all distinct label sequences seen so far. At inference, the most relevant label sequence will be retrieved from $\mathcal{P}$ based on the similarity between all the candidate labels and y_{gen} generated by $\mathcal{M}$ given the input x:

$$\begin{aligned} y_{\text{gen}} &= \text{generate}(\mathcal{M}, x) \\ y_{\text{pred}} &= \operatorname*{argmax}_{y \in \mathcal{P}} \cos(\text{embed}(y), \text{embed}(y_{\text{gen}})) \end{aligned} \quad (2)$$

Here, $\text{embed}(\cdot)$ is parameterized by a Sentence-BERT model (Reimers and Gurevych, 2019).

Although the idea of solving CIL via generation is simple, the framework change yields a great

[3] Readers can refer to Appendix A for more **related works**.

performance boost. Figure 2 compares the classifier framework and the generation framework on CLINC150 (Larson et al., 2019) which contains 150 classes and is split into 15 tasks. With no additional mechanism to handle CF, using the same PLM, *i.e.* BART$_{\text{base}}$ (Lewis et al., 2020), the generation framework gives much better results.

Generation loss prevents PLMs from collapsing. To understand the inner working of the framework change, we look into the PLM's representation ability in the CIL process. Unlike single-task learning, CIL requires the PLM to maintain the representation ability as much as possible for future classes, which is nontrivial because PLMs tend to have representation collapse[4] during fine-tuning (Aghajanyan et al., 2021). Figure 2 (b) compares the change of the PLM's representation ability in the two frameworks by using the neural collapse metric ($\mathcal{NC}$) proposed in Zhu et al. (2021c):

$$\mathcal{NC} := \frac{1}{K}\operatorname{trace}\left(\mathbf{\Sigma}_W \mathbf{\Sigma}_B^{\dagger}\right), \qquad (3)$$

where $\mathbf{\Sigma}_W, \mathbf{\Sigma}_B \in \mathbb{R}^{d_{\text{enc}} \times d_{\text{enc}}}$ denote the within-class and between-class covariance matrices of the encoded sequences, $\mathbf{\Sigma}_B^{\dagger}$ denotes the pseudo inverse of $\mathbf{\Sigma}_B$, and K denotes the number of classes in the dataset. As clearly shown, when learning more and more tasks, both frameworks witness a drop of the PLM's representation ability. However, the PLM in the generation framework keeps a relatively steady representation ability in the CIL process, thus remaining capable of learning unseen classes.

3.2 Vocabulary-Aware Generation Loss

One major challenge of CIL is that the previously learned decision boundaries may be corrupted when the model weights are updated to learn new classes (Zhu et al., 2021a). Beyond using the generation framework to retain the PLM's representation ability, we further propose a *vocabulary-aware generation loss* (VAG loss) to ease the task interference (which causes catastrophic forgetting).

Note that although the PLM is pre-trained with a large vocabulary (*e.g.*, BART has a vocabulary size of 50,265), only a tiny subset will be used for the label generation in each task. VAG loss leverages this natural sparsity of vocabulary by masking the probability of tokens that will not be used in the current task before calculating the generation loss.

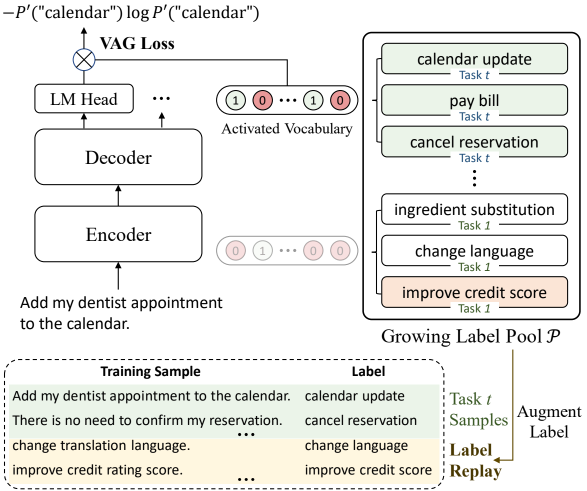

Figure 3: Overview of training VAG on task t. VAG modifies the generation loss by masking the probability of unused vocabulary and creates pseudo replay data by augmenting the label sequences.

Specifically, denote the vocabulary set of $\mathcal{C}_t$ as $\mathcal{V}_t$, $P(y_i|c, Y_{1:i-1})$ in Equation (1) is changed to

$$P'(y_i|c, Y_{1:i-1}) = \frac{\exp(E_{y_i}^{\mathsf{T}} f_{\theta_{dec}}(c, Y_{1:i-1}))}{\sum_{w \in \mathcal{V}_t} \exp(E_w^{\mathsf{T}} f_{\theta_{dec}}(c, Y_{1:i-1}))}. \qquad (4)$$

Since $|\mathcal{V}_t| \ll |\mathcal{V}|$, maximizing the modified probability leads to a sparse update of E and effectively eases the forgetting of previous classes.

3.3 Label-based Pseudo Replay

Another major challenge of CIL is that the system needs to separate new classes in task t and classes in previous tasks since the task identity is unknown at inference. To help construct decision boundaries across tasks and mitigate forgetting, VAG creates pseudo replay data by *augmenting the label sequences* in previous tasks.

Specifically, given the label sequence y, the augmented sequence $\text{aug}(y)$ will be used as a pseudo replay data instance with label y. To preserve the label semantics as well as to create diverse samples, we implement $\text{aug}(\cdot)$ by randomly adding related tokens to the original label sequence based on contextual word embeddings (Ma, 2019):

$$\mathcal{D}_{<t}^{LPR} = \{(\text{aug}(y), y) | y \in \cup_{i=1}^{t-1} \mathcal{Y}_i\} \qquad (5)$$

When training task t, we sample $\lambda|\mathcal{D}_t|$ pairs from $\mathcal{D}_{<t}^{LPR}$ (λ is a hyper-parameter), and combine them with $\mathcal{D}_t$ as the training data. The VAG loss is also applied to the pseudo replay sample $(\text{aug}(y), y)$, *i.e.*, for each $y \in \mathcal{Y}_i$, its associated vocabulary subset $\mathcal{V}_i$ will be used in the denominator in Equation (4).

[4] Representation collapse refers to the degradation of generalizable representations of pre-trained models during fine-tuning (Aghajanyan et al., 2021).

4 Experiments

4.1 Datasets and Baselines

Datasets. We use 5 datasets. Following Wu et al. (2022), we randomly split each dataset into X tasks with Y classes per task, expressed as (X/Y). CLINC150 (Larson et al., 2019) (15/10) and Banking77 (Casanueva et al., 2020) (7/10) for intent classification, 20 Newsgroups (20News) (Lang, 1995) (10/2) for topic classification, FewRel (Han et al., 2018) (8/10) and TACRED (Zhang et al., 2017) (8/5) for relation classification. Additional details about the datasets are given in Appendix B.1.

Baselines. We consider the following baselines: (1) **Vanilla** fine-tunes the PLM sequentially. (2) **EWC** (Kirkpatrick et al., 2017) is a regularization-based method. (3) **KD** (Hinton et al., 2015) uses knowledge distillation. (4) **L2P** (Wang et al., 2022) dynamically prompts the PLM without the task identity. These baselines use the classifier framework, and we adapt them to the generation framework as another set of baselines (**X-G**). We also consider 3 methods which use generation for CL: (5) **LAMOL** (Sun et al., 2019) fine-tunes GPT-2 continually with manual prompts and incorporates pseudo replay. Since LAMOL is a TIL system, we adapt it to CIL by using the same prompt. (6) **PAGeR** (Varshney et al., 2022) extends LAMOL with contrastive training and knowledge distillation. (7) **ACM** (Zhang et al., 2022) extends LAMOL by adding compositional adapters. ACM is not designed for classification, so we adapt it by training the PLM to generate the class label.

Implementation details are in Appendix B.2.

4.2 Main Results

Table 1 shows the results in the non-exemplar (non-replay) based CIL setting. The reported results are averaged over 5 random seeds.

Baselines using the generation objective give better results. In accord with the findings in Wu et al. (2022), regularization-based methods (*e.g.*, EWC, KD) perform poorly. For L2P, although it keeps the PLM fixed, the algorithm cannot converge in our experiments due to the randomness introduced by the error-prone prompt selection. Comparing the same method in two frameworks (*e.g.*, EWC *v.s.* EWC-G), we can see that the framework switch is highly effective, which indicates the superiority of solving CIL via label generation. Moreover, the best-performing baseline ACM also adopts the generation objective.

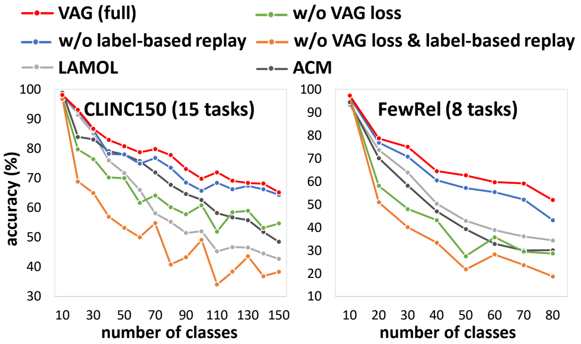

Figure 4: Changes in accuracy (%) with increasing tasks through the class-incremental learning process.

Superiority of VAG. On all the datasets, VAG achieves the best performance, even outperforming other baselines in the generation framework by a large margin (Table 1). Figure 4 also shows that VAG has less forgetting in the CIL process than the two best baselines. However, compared with the results in the non-continual learning setting (Non-CL in Table 1) which represent the performance upper bound for each dataset, our method still has considerable room for improvement, thereby encouraging future endeavors.

Extending VAG to use real replay data. Notably, VAG can be directly extended to utilize real or saved replay data when they are available. Since real replay data are from the training distribution, we optimize the original generation loss upon the combination of $\mathcal{D}_t$ and the real replay data besides optimizing the VAG loss.[5] We consider **ER** (Lopez-Paz and Ranzato, 2017), **DER++** (Buzzega et al., 2020) and **LDBR** (Huang et al., 2021) as replay-based baselines and experiment with different replay buffer sizes. Table 2 shows the comparison results. VAG still performs the best, especially when the buffer size is small (see the *Avg.* row)[6].

4.3 Ablation Study and Analysis

We analyze the effect of each component in our VAG system and Figure 4 shows the ablation results. While the full VAG uniformly gives the best results, we further observe that: (1) Both VAG loss and label-based replay can benefit CIL independently. (2) Label-based replay has a relatively small effect especially when we have already adopted VAG loss.

[5] More details are included in Appendix B.3.

[6] When the buffer size is large, all the methods approach the non-CL results (performance upper bound), so the performance gap between VAG and other baselines gets smaller.

	#Tasks	Softmax Classifier				Generation								
		Vanilla	EWC	KD	L2P	Vanilla-G	EWC-G	KD-G	L2P-G	LAMOL	PAGeR	ACM	**VAG**	Non-CL
CLINC150	15	7.37	7.67	9.39	3.32	37.63	44.23	36.51	43.84	42.56	39.39	48.78	**65.69**	94.66
Banking77	7	14.43	14.51	14.59	1.98	26.88	29.99	21.36	34.42	39.51	43.85	54.72	**55.19**	88.61
20News	10	9.96	9.96	10.00	6.84	44.17	49.81	30.84	25.47	52.05	49.61	60.79	**73.51**	86.81
FewRel	8	12.39	13.09	12.33	6.60	19.44	25.12	15.95	6.52	34.69	39.09	29.74	**52.26**	85.14
TACRED	8	10.96	10.41	12.04	4.85	23.44	24.36	17.44	10.18	16.46	27.99	18.67	**46.15**	70.38
Avg.	\	11.02	11.13	11.67	4.72	30.31	34.70	24.42	24.09	37.05	39.99	42.54	**58.56**	85.12

Table 1: Final accuracy (%) of VAG and baseline methods for non-exemplar based CIL. The gray column shows the results in the non-continual learning setting which provides an upper bound. The reported results are averaged over 5 random seeds and the **standard deviations** are reported in Appendix B.4.

	Ours (non-exemplar)	Buffer size = 1%				Buffer size = 3%				Buffer size = 5%			
		ER	DER++	LDBR	**VAG**	ER	DER++	LDBR	**VAG**	ER	DER++	LDBR	**VAG**
CLINC150	65.69	55.62	56.85	67.34	**72.44**	78.06	73.29	81.34	**81.53**	85.31	80.37	**86.49**	85.00
Banking77	55.19	45.24	48.32	54.76	**58.96**	65.22	65.73	70.16	**70.57**	74.32	73.06	74.37	**74.81**
20News	73.51	84.53	84.24	**85.30**	84.76	85.45	85.30	**86.53**	85.29	85.79	85.66	**86.83**	85.85
FewRel	52.26	60.77	63.21	51.26	**68.56**	74.20	72.92	65.21	**75.99**	78.08	78.09	70.48	**78.42**
TACRED	46.15	36.09	37.03	38.21	**49.70**	49.66	52.12	46.93	**58.00**	56.93	55.72	52.22	**61.28**
Avg.	58.56	56.45	57.93	59.37	**66.88**	70.52	69.87	70.03	**74.28**	76.09	74.58	74.08	**77.07**

Table 2: Final accuracy (%) of VAG and exemplar-based baselines for CIL with different buffer sizes (*i.e.*, we save 1%, 3%, 5% of previous training data). The **standard deviations** are reported in Appendix B.4.

In Appendix C, we compare the confusion matrices of "VAG (full)" and "w/o VAG loss". We find VAG loss effectively prevents the model from biasing towards predicting the latest learned classes, thus effectively easing the forgetting issue. In Appendix D, we further analyze the impact of different label-based replay ratios (λ in §3.3). Figure 6 shows that a small amount of label-based replay data already improves the results markedly, indicating the usefulness of leveraging label semantics for pseudo replay.

As discussed in §3.1, the generation loss eases the drop of the PLM's representation power in the CIL process. Appendix E reports the neural collapse metric $\mathcal{NC}$ of different methods after CIL. The VAG system preserves the representation ability of the PLM to the greatest extent.

5 Conclusion

We presented the VAG system which solves CIL based on label generation. We showed that migrating to the generation framework gives a drastic performance boost and eases the representation collapse of the pre-trained model. Experimental results demonstrate the effectiveness of VAG.

Limitations

One limitation of this work is that VAG does not achieve zero forgetting. Although we show solving CIL based on label generation can effectively ease forgetting and representation collapse of the pre-trained model, it is still interesting to further explore how to explicitly solve the forgetting issue in this new framework. The proposed techniques in VAG are a step in the exploration.

Another limitation is that we directly use the label sequences provided by the original dataset. This may be suboptimal because the quality of the manually created label is hard to guarantee as it may fail to capture the semantic information of the samples in a class. A potential direction is to study creating label sequences automatically by summarizing the training samples. We leave this for future work.

Ethics Statement

While our proposed VAG system involves generation, it does not have the general ethical concern of generation, *i.e.*, outputting biased or discriminative texts, because the final output of the system is retrieved from the label pool which is highly controllable. For our experiments, we use public datasets and believe that none of them contains offensive contents. Also, although the training of the VAG system requires computational resources, the CIL paradigm is resource-efficient because the model preserves the previously learned knowledge and continually learns new classes.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440, Online. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2736–2746, Online. Association for Computational Linguistics.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021. CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6871–6883, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. 2022. A theoretical study on solving continual learning. In *Advances in Neural Information Processing Systems*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Guodun Li, Yuchen Zhai, Qianglong Chen, Xing Gao, Ji Zhang, and Yin Zhang. 2022. Continual few-shot intent detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 333–343, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. 2021. Lifelong intent detection via multi-strategy rebalancing. *arXiv preprint arXiv:2108.04445*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3461–3474, Online. Association for Computational Linguistics.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13570–13577.

Ramakanth Pasunuru, Veselin Stoyanov, and Mohit Bansal. 2021. Continual few-shot learning for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5688–5702, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chengwei Qin and Shafiq Joty. 2022. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Vaibhav Varshney, Mayur Patidar, Rajat Kumar, Lovekesh Vig, and Gautam Shroff. 2022. Prompt augmented generative replay via supervised contrastive learning for lifelong intent detection. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1113–1127, Seattle, United States. Association for Computational Linguistics.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. 2020. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*.

Yu Xia, Quan Wang, Yajuan Lyu, Yong Zhu, Wenhao Wu, Sujian Li, and Dai Dai. 2022. Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2291–2300, Dublin, Ireland. Association for Computational Linguistics.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3062–3072, Dublin, Ireland. Association for Computational Linguistics.

Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual sequence generation with adaptive compositional modules. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3653–3667, Dublin, Ireland. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Kang Zhao, Hua Xu, Jiangong Yang, and Kai Gao. 2022. Consistent representation learning for continual relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3402–3411, Dublin, Ireland. Association for Computational Linguistics.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. 2021a. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34:14306–14318.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. 2021b. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. 2021c. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834.

A Related Work

Continual Learning. Continual learning requires a model to sequentially learn a series of tasks. The main challenge that existing papers focus on is overcoming *catastrophic forgetting* (CF) (McCloskey and Cohen, 1989). Previous works usually fall in the following categories: (1) Regularization-based methods, which penalize the parameter update and preserve the previous task knowledge (Kirkpatrick et al., 2017; Huang et al., 2021; Zhu et al., 2021b; Li and Hoiem, 2017). (2) Parameter-isolation methods, which separate parameters for different tasks by finding subnetworks in the over-parameterized model (Wortsman et al., 2020; Serra et al., 2018; Mallya and Lazebnik, 2018) or adding additional task-specific modules (Houlsby et al., 2019; Ke et al., 2021). These methods need to know the task identity for inference. (3) Replay-based methods, which jointly train the model with new task data and some saved examples (Lopez-Paz and Ranzato, 2017; Buzzega et al., 2020) or generated pseudo data (Shin et al., 2017; Sun et al., 2019) of previous tasks. In real applications, storing replay samples may not be possible due to the data privacy issue or memory overhead (Zhu et al., 2021b).

Based on the differences in evaluation protocols, continual learning can be summarized into three major settings: class-incremental learning (CIL), task-incremental learning (TIL), and domain-incremental learning (DIL) (Yin et al., 2022). Among them, CIL which aims to build a single predictive model on all seen classes, is the most difficult one because the task identity is not available for inference. This requires the model to not only tackle catastrophic forgetting of the within-task prediction ability but also predict the task identity correctly (Kim et al., 2022). In the language domain, prior works have studied CIL for intent detection (Liu et al., 2021; Li et al., 2022), relation classification (Han et al., 2020; Zhao et al., 2022), named entity recognition (Monaikul et al., 2021; Xia et al., 2022), *etc.* Despite the great success of pre-trained language models (PLMs), these models still suffer from severe CF issue in continual learning. In a large-scale comparative study, Wu et al. (2022) concluded that PLMs perform extremely poorly in the CIL setting. In their study, a PLM is leveraged by fine-tuning the model with a classification head. However, in this work, we find that PLMs can show better CIL ability if we fine-tune the PLM in a generation framework.

Dataset	Class	Task	Train	Validation	Test
CLINC150	150	15	15,000	3,000	4,500
Banking77	77	7	7,191	1,800	2,800
20News	20	10	10,000	3,998	5,999
FewRel	80	8	33,600	11,200	11,200
TACRED	42	8	5,909	1,482	1,259

Table 3: Dataset statistics. Banking77 and TACRED do not have the validation set, so we randomly sample 20% data from the training set for validation.

Text Generation in Continual Learning Study. With the success of natural language generation using PLMs (Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020), some works on continual learning of NLP utilize the generation ability of PLMs to unify different potential tasks through prompting (Qin and Joty, 2022) or instruction tuning (Yin et al., 2022; Scialom et al., 2022). The text generation can also be used to create pseudo replay data for previous task. LAMOL (Sun et al., 2019) is a typical system in this line of work which simultaneously learns to solve all the tasks in a unified question-answering manner and generates pseudo replay samples in the TIL setting. While LAMOL is closely related to our work which also leverages generation, the key difference is that we focus on CIL instead of TIL and show for the first time that the generation objective itself can effectively ease the CF issue. We also show that the generation objective bears a link with preventing the representation collapse of the PLM and further propose the VAG approach to exploit the generation framework for CIL. Some other works in the continual learning literature directly focus on generation tasks (not classification tasks) and study the problem of continual sequence generation (Zhang et al., 2022; Mi et al., 2020). These works naturally involve generation due to the property of their studied tasks.

B Additional Details of Experiments

B.1 Dataset Details

As described in §4.1, we use 5 datasets for our experiments. **CLINC150** (Larson et al., 2019) and **Banking77** (Casanueva et al., 2020) are two intent classification datasets with 150 classes and 77 classes respectively. Each intent class is described by a short phrase (*e.g.*, “change language”, “edit personal details”) in the original dataset, and we directly use these phrases as the label sequences.

20 Newsgroups (20News) is a topic classification dataset with 20 categories associated with hierarchical labels (*e.g.*, "comp.sys.ibm.pc.hardware" and "misc.forsale"). We convert the hierarchical labels into label sequences by replacing "." with a whitespace and extending the abbreviations into complete words (*e.g.*, "computer system ibm pc hardware", "miscellaneous forsale"). **FewRel** (Han et al., 2018) is a relation classification dataset with 80 relations. **TACRED** (Zhang et al., 2017) is another relation classification dataset with 42 relations and it has highly unbalanced samples for each relation. In these two datasets, each relation is described by a short phrase (*e.g.*, "exhibition history", "organization related: founded by") and we use them as the label sequences.

Following Wu et al. (2022), we randomly split CLINC150, Banking77, FewRel into disjoint tasks with 10 classes per task. We split 20News into 10 tasks with 2 classes per task and TACRED into 8 tasks with 5 classes per task for a more challenging evaluation. Table 3 summarizes the dataset statistics.

Note that among the datasets we used, CLINC150[7], Banking77[8], FewRel[9], TACRED[10] are licensed. We ensure that we did not violate any license condition when conducting our experiments.

B.2 Implementation Details

We implement VAG and baseline (1)-(4) with the Transformers library (Wolf et al., 2020) and use BART$_{\text{base}}$[11] (#parameters: 139M) as the backbone PLM. For LAMOL[12] and ACM[13], we directly use their official implementation and use the same question prompt for each task[14] so that they do not need the task identity for inference any more and can suit the CIL setting. For PAGeR, we use our own implementation because its source code is not publicly available. Table 4 gives the hyper-parameters of baseline implementations.

For learning each task, we train the model for 10 epochs and use the validation set of the current task for early stopping. We set the batch size as 8 and the max sequence length as 128. We use AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the learning rate of 1e-5. For the label-based pseudo replay component of VAG, we implement $\text{aug}(\cdot)$ using the `ContextualWordEmbdsAug` in the nlpaug library[15] which adds $0.3 \times \text{token\_num}(y)$ related tokens to the original label sequence y and the hyper-parameter λ is set to 0.1. At inference, we use greedy decoding to decode the generated sequence and $\text{embed}(\cdot)$ in Equation (2) is parameterized by `paraphrase-MiniLM-L6-v2` provided in the Sentence-Transformers library[16]. We use NVIDIA GeForce RTX 2080 Ti GPU to conduct all our experiments.

B.3 Exemplar-Based Setting

As discussed in §4.2, we extend VAG system to the exemplar-based CIL setting where real replay data are available. In exemplar-based CIL, the training objective of VAG at task *t* is to minimize

$$\mathbb{E}_{\mathcal{D}^{ER}_{<t}\cup\mathcal{D}_t}[\ell_{normal}(x,y)] + \mu\mathbb{E}_{\mathcal{D}^{LPR}_{<t}\cup\mathcal{D}_t}[\ell_{VAG}(x,y)], \tag{6}$$

where $\mathcal{D}^{ER}_{<t}$ represents the real replay data of previous tasks, $\mathcal{D}^{LPR}_{<t}$ represents the label-based pseudo replay data (see Equation (5)), and μ is a hyper-parameter balancing two replay terms. We set μ to 1 in our experiments.

For comparison, we consider 3 typical replay-based methods: (1) **ER** (Lopez-Paz and Ranzato, 2017) directly combines replay samples and current task samples in training batches to fine-tune the classifier. (2) **DER++** (Buzzega et al., 2020) exploits replay data in training and adds a regularization term to prevent the logits of replay data from changing. (3) **LDBR** (Huang et al., 2021) uses information disentanglement based regularization and selects replay samples through K-means clustering. We experiment with different buffer sizes by storing 1%, 3%, and 5% of previous training data. Other training hyper-parameters are in accord with the non-exemplar based setting.

[7] https://github.com/clinc/oos-eval/blob/master/LICENSE

[8] https://github.com/PolyAI-LDN/task-specific-datasets/blob/master/LICENSE

[9] https://github.com/thunlp/FewRel/blob/master/LICENSE

[10] https://catalog.ldc.upenn.edu/LDC2018T24

[11] https://huggingface.co/facebook/bart-base

[12] https://github.com/chho33/LAMOL

[13] https://github.com/SALT-NLP/Adaptive-Compositional-Modules

[14] For CLINC150, Banking77, 20News, we set the question prompt to be "What's the category of this text?". For FewRel and TACRED, we set the question prompt to be "What's the relation between these two entities?".

[15] https://pypi.org/project/nlpaug/

[16] huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2

Method	Key	Value	Note
EWC	λ	5,000	The weight for penalty, selected from [500, 1,000, 2,000, 5,000, 10,000, 20,000, 50,000].
KD	λ	0.1	The weight for knowledge distillation loss, selected from [0.1, 0.5, 1.0].
L2P	M	10	The total number of prompts, following the original paper.
	N	5	The number of dynamically selected prompts, following the original paper.
	λ	0.5	The weight of key selection loss, following the original paper.
LAMOL	γ	0.2	The sampling ratio of pseudo replay data, following the original paper.
PAGeR	λ_1	1	The weight of the generation loss and distillation loss, following the original paper.
	λ_2	0.25	The weight of the replay data generation loss, following the original paper.
	λ_3	0.25	The weight of the supervised contrastive training loss, following the original paper.
	γ	0.2	Refer to γ in LAMOL.
ACM	γ	0.01	The entropy coefficient, using the default value of the official implementation.
	c	0.15	The initialization of the coefficient weights, using the default value of the official implementation.

Table 4: The hyper-parameters of baseline implementation.

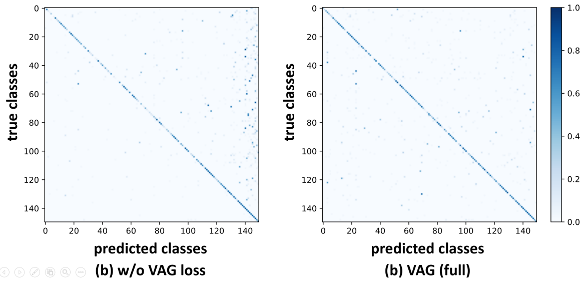

Figure 5: Confusion matrix of “VAG (full)” and “w/o VAG loss” on CLINC150 (15 tasks).

B.4 Standard Deviations

In §4.2, we evaluated our proposed system VAG in both non-exemplar and exemplar-based CIL setting. Table 5 and Table 6 give the standard deviations of the reported results.

C Confusion Matrices

In §4.3, we analyze the effectiveness of each component in the proposed VAG system. To study the effect of VAG loss, we compare the confusion matrixes of “VAG (full)” and “w/o VAG loss”. As shown in Figure 5, VAG loss effectively prevents the model from having a strong bias towards predicting the latest learned classes. Since VAG loss limits the denominator to the vocabulary used by the current task, training with VAG loss has less interference to previous task knowledge, thus yielding better final performance.

D Analysis of Label-Based Replay Ratio

As discussed in §3.3, VAG samples $\lambda|\mathcal{D}_t|$ pseudo replay data instances created by label-based data augmentation and combines them with $\mathcal{D}_t$ as the training data. Here, we analyze the impact of different label-based replay ratios λ. Figure 6 shows the results. We observe that a small amount of label-based replay data can already yield improvements and the results are similar when we further increase the label-based replay ratio λ. We set λ to 0.1 in our main experiments (see §4).

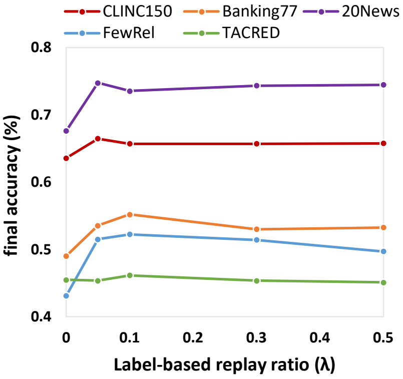

Figure 6: Final results (accuracy) with different label-based replay ratios.

E Neural Collapse with Different Methods

As discussed in §3.1, we find the generation framework can better preserve the representation ability of the pre-trained model in the CIL process. Table 7 gives the neural collapse metric $\mathcal{NC}$ of different methods after CIL. In general, after the continual learning process, all the models have lower $\mathcal{NC}$ compared with the original PLM, especially when we fine-tuned the PLM using the traditional classifier framework. We also observe that while we modify the generation loss in the VAG system, its desired property is retained and our proposed CIL

	Softmax Classifier				Generation								
	Vanilla	EWC	KD	L2P	Vanilla-G	EWC-G	KD-G	L2P-G	LAMOL	PAGeR	ACM	**VAG**	Non-CL
CLINC150	±0.56	±0.50	±1.50	±0.34	±2.95	±1.72	±1.44	±4.99	±0.74	±3.04	±2.50	±1.54	±0.67
Banking77	±0.68	±0.51	±0.46	±0.40	±3.28	±2.02	±0.83	±3.01	±0.92	±2.78	±1.54	±0.37	±0.94
20News	±0.02	±0.01	±0.04	±0.35	±3.43	±5.04	±2.02	±1.69	±2.80	±1.55	±2.55	±3.81	±0.35
FewRel	±0.30	±0.55	±1.06	±0.68	±1.26	±1.14	±1.13	±3.43	±1.41	±1.69	±1.88	±1.29	±0.73
TACRED	±1.09	±0.29	±1.33	±0.30	±1.08	±1.36	±1.30	±0.94	±0.26	±1.08	±1.76	±0.59	±0.33

Table 5: Standard deviations of the proposed VAG system and the baselines in non-exemplar based class-incremental learning setting. The corresponding averaged results are in Table 1.

	VAG (non-exemplar)	**Buffer size = 1%**				**Buffer size = 3%**				**Buffer size = 5%**			
		ER	DER++	LDBR	**VAG**	ER	DER++	LDBR	**VAG**	ER	DER++	LDBR	**VAG**
CLINC150	±1.54	±8.42	±7.90	±1.75	±0.56	±3.35	±0.83	±1.52	±0.88	±1.40	±1.17	±0.50	±1.05
Banking77	±0.37	±6.38	±2.78	±1.80	±1.95	±2.24	±1.21	±0.09	±1.72	±2.77	±1.69	±2.48	±1.18
20News	±3.81	±1.01	±1.41	±0.04	±0.39	±0.28	±0.28	±0.35	±0.49	±0.28	±0.07	±0.34	±0.28
FewRel	±1.29	±3.37	±4.91	±1.46	±0.94	±0.92	±1.41	±1.41	±0.65	±0.72	±1.21	±1.74	±0.63
TACRED	±0.59	±3.85	±3.97	±0.71	±2.02	±2.61	±4.30	±1.47	±3.24	±2.96	±1.75	±1.43	±0.99

Table 6: Standard deviations of the proposed VAG system and the baselines for class-incremental learning setting with different buffer sizes. The corresponding averaged results are in Table 2.

	PLM (before CIL)	Vanilla	Vanilla-G	**VAG**
CLINC150	65.84	8.70	53.47	**57.24**
Banking77	109.55	46.34	**72.34**	71.04
20News	15.92	2.16	13.95	**15.51**
FewRel	321.09	77.31	170.25	**190.09**
TACRED	46.79	32.78	40.54	**45.54**

Table 7: $\mathcal{NC}$ of models before and after class-incremental learning with different training methods.

framework preserves the representation ability of the PLM to the greatest extent.