LSEBMCL: A Latent Space Energy-Based Model for Continual Learning

Xiaodi Li
Department of Electrical and Computer Engineering
The University of Texas at Dallas
Richardson, USA
xiaodi.li@utdallas.edu

Dingcheng Li
Vextex AI
Google
Kirkland, USA
dingchengli@google.com

Rujun Gao
Department of Mechanical Engineering
Texas A&M University
College Station, USA
grj1214@tamu.edu

Mahmoud Zamani
Department of Electrical and Computer Engineering
The University of Texas at Dallas
Richardson, USA
mxz173130@utdallas.edu

Latifur Khan
Computer Science Department
The University of Texas at Dallas
Richardson, USA
lkhan@utdallas.edu

***Abstract*—Continual learning has become essential in many practical applications such as online news summaries and product classification. The primary challenge is known as catastrophic forgetting, a phenomenon where a model inadvertently discards previously learned knowledge when it is trained on new tasks. Existing solutions involve storing exemplars from previous classes, regularizing parameters during the fine-tuning process, or assigning different model parameters to each task. The proposed solution LSEBMCL (Latent Space Energy-Based Model for Continual Learning) in this work is to use energy-based models (EBMs) to prevent catastrophic forgetting by sampling data points from previous tasks when training on new ones. The EBM is a machine learning model that associates an energy value with each input data point. The proposed method uses an EBM layer as an outer-generator in the continual learning framework for NLP tasks. The study demonstrates the efficacy of EBM in NLP tasks, achieving state-of-the-art results in all experiments.**

***Index Terms*—continual learning, energy-based model, catastrophic forgetting, question answering, language generation**

I. Introduction

Label prediction for continuously occurring instances is crucial in practical applications like online news summaries, product classification, and dialogue learning systems. To address these scenarios, models must acquire, fine-tune, and transfer knowledge over time, a concept referred to as continual learning [1]. Continual Learning (CL) aims to create systems that can rapidly acquire new skills and integrate them with prior knowledge, mimicking human learning. A key challenge in CL is catastrophic forgetting, where models forget previously learned knowledge when training on new tasks.

Approaches to mitigate catastrophic forgetting can be categorized into: (1) storing exemplars from previous tasks; (2) parameter regularization during fine-tuning; and (3) task-specific parameter allocation. These methods aim to retain prior knowledge while learning new tasks. Our method prevents forgetting by sampling data from previous tasks using an Energy-based Model (EBM) during training. The EBM is first trained on each task, enabling the retention of prior knowledge and improving performance on subsequent tasks.

An EBM associates scalar energy values with input data points, assigning lower energy to more likely inputs and higher energy to less likely ones. EBMs are applicable to various tasks, including classification and regression. For instance, Pang et al. [2] used EBMs in the latent space to enhance the expressivity of generative models, demonstrating its utility in improving latent space structure for both generation and classification.

Despite advancements in CL, EBM applications in this domain remain limited. For example, [3] applied EBMs for classification in complex CL scenarios like boundary-agnostic and class-incremental learning. Unlike their approach, which uses EBMs as the core model, we employ EBMs as an external sampling mechanism, simplifying updates and preserving flexibility. Additionally, while their work focuses on computer vision datasets, our study applies EBMs to natural language processing datasets.

Our proposed method, LSEBMCL, makes three key contributions: (1) integrating an EBM layer into a continual learning framework for NLP tasks for the first time; (2) addressing catastrophic forgetting in NLP tasks using EBM; and (3) achieving state-of-the-art performance across experiments, demonstrating the effectiveness of our approach.

II. Related Works

A. Continual Learning

Continual learning involves sequential tasks and is particularly relevant in scenarios where data arrives in a non-i.i.d. manner and new tasks emerge. However, deep neural networks face the challenge of *catastrophic forgetting*, which hinders their ability to retain prior knowledge. Current continual learning methods fall into three categories: (1) Replay methods [4], [5]; (2) Regularization-based methods [6], [7]; and (3) Parameter isolation methods [8], [9]. Replay methods either

store raw samples or generate pseudo-samples using generative models. For instance, iCaRL [10] stores class exemplars, and GEM [11] uses gradients from previous tasks to constrain updates. LAMOL [12] reduces forgetting by generating artificial examples of previous tasks. Regularization-based methods avoid storing raw inputs, introducing regularization terms to consolidate knowledge. LwF [13] uses model outputs as soft labels, while MAS [14] estimates parameter importance for adaptation. IDBR [15] applies disentanglement-based regularization for text classification, and LPC [16] combines parameter calibration with logit preservation. Parameter isolation methods allocate distinct parameters per task to prevent forgetting, as seen in PackNet [17] and HAT [18]. Our method employs replay-based EBMs to generate artificial samples for previous tasks.

B. Energy-based Models

Energy-based models (EBMs) represent probability density functions via an energy function $E(x)$, mapping realistic points to low energy values and unrealistic points to high values [19]. EBMs offer simplicity, stability, and parameter efficiency. Recent advancements enable EBMs to model high-dimensional data [20], [21], and latent space EBMs [2] improve model expressivity for tasks such as text generation and trajectory modeling. EBMs have been shown to prevent catastrophic forgetting in continual learning [3]. Unlike [3], which uses EBMs as the primary model for classification, our method employs EBMs as an outer-generator for tasks such as classification and text generation. Other applications of EBMs include joint training with pretrained text encoders to enhance calibration [22] and leveraging low-dimensional structures for anomaly detection [23].

III. Methodology

We initiate the process with a pre-trained base model, specifically the latest Large Language Model (LLM) Mistral 7B. [24]. We have developed the LSEBMCL model, which consists of four main components. The initial component is the Inference Network, followed by Operator 1, Operator 2, and finally, the Energy Function. In the subsequent sections, we will provide a detailed introduction to each of these components individually.

A. Inference Network

Following decaNLP [25], we make the datasets pre-processed as a QA (Question Answering) problem. First, we convert continual learning tasks to a unified QA format as:

$$< x, y >\in D \tag{1}$$

where x is the Question, y is the Answer, and D is a training set of QA. We target handling diverse tasks, covering Question Answering (QA), Natural Language Inference (NLI), Sentiment Analysis (SA), Semantic Role Labeling (SRL), etc.

We introduce an inference network, denoted as $A_\Psi(x)$, which is also referred to as the "energy-based inference network" (as depicted at the bottom of Figure 1). This network is parameterized by Ψ and trained with the objective of achieving the following goal:

$$A_\Psi(x) \approx argmin_{y \in Y_R(x)} E_\Theta(x, y) \tag{2}$$

where $Y_R(x)$ are the true labels. Specifically, we train the inference network parameters Ψ as follows (assuming Θ is the parameters of EBM):

$$\widehat{\Psi} = argmin_\Psi \sum_{<x,y>\in D} E_\Theta(x, A_\Psi(x)) \tag{3}$$

B. Operator 1

We employ two operators o^1 and o^2 that are used to map z_t logits into distributions for use in the energy. As shown at the middle of Figure 1, we seek an operator o^1 to modulate the way that logits z_t output by the inference network is fed to the decoder input slots in the energy function. o^1 is the operation for feeding inference network outputs into the decoder input slots in the energy.

C. Operator 2

An operator o^2 to determine how the distribution $p_\Theta(\cdot|...)$ is used to compute the log probability of y. o^2 is the operation for computing the energy on the output. Explicitly, then, we write each local energy term as

$$e_m(x, y) = -o^2(z_m) \log p_\Theta(\cdot|o^1(z_m), x_m) \tag{4}$$

Our objective is to minimize the aforementioned energy function concerning the variable z_t in our inference networks. The softmax operation is selected for o^1 and o^2.

D. Energy Function

Figure 1 shows the proposed latent space EBM continual learning model. We design an EBM layer to play the role of outer-generator. After training each task, the EBM model generates samples based on data from previous tasks before training on the new task. During training on the new task, the model not only trains a model on the new data but also trains on the extra data generated from previous tasks by the EBM model as shown at the top of Figure 1. The equation is shown below:

$$E_\Theta(x, y) = \sum_{m=1}^{M} e_m(x, y) \tag{5}$$

where Θ is the parameters of the EBM. m means the index of the task and M means the total number of tasks. $e_m(x, y)$ is calculated using the following equation:

$$e_m(x, y) = -\log p(y_m|x_m) \tag{6}$$

In Figure 1, for each task, we have:

$$p_\theta(x, z) = p_\alpha(z) p_\beta(x|z) \tag{7}$$

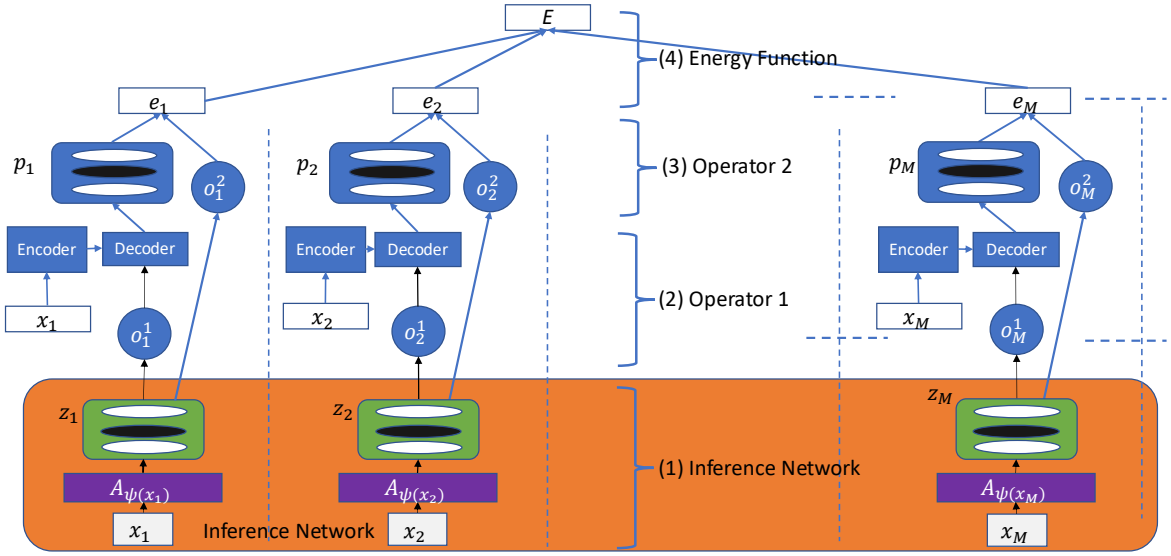

Fig. 1. The Overview of LSEBMCL Framework. (1) Inference Network: The process begins with the inference network at the bottom, where inputs (x) are processed to generate encoded representations (z). (2) Operator 1 and (3) Operator 2: These operators facilitate the transition of logits from the inference network to the decoder inputs and compute the energy on the outputs, respectively. (4) Energy Function: At the culmination of the process, the energy function evaluates the outputs, contributing to the model's generation.

where $p_\alpha(z)$ is the prior model with parameters α, z is a latent dense continuous vector, and $p_\beta(x|z)$ is given by a generative model parameterized with β.

The EBM prior of z, $p_\alpha(z)$, is in the form of the energy-based correction of an isotropic Gaussian reference distribution $p_0(z)$:

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp[F_\alpha(z)] p_0(z) \propto \exp[F_\alpha(z) - \frac{1}{2\sigma^2}\|z\|^2] = \exp E(z) \tag{8}$$

where $E(z) = F_\alpha(z) - \frac{1}{2\sigma^2}\|z\|^2$ is the energy function that maps z to a scalar, where $F_\alpha(z)$ is parameterized by a small multi-layer perceptron (MLP), σ^2 is the regularization related hyper-parameter. $Z(\alpha) = \int \exp[F_\alpha(z)] p_0(z) dz$ is the normalizing constant/partition function.

For language modeling, assuming $x \in R^D$,

$$x = g_\beta(z) + \epsilon \tag{9}$$

where $\epsilon \sim N(0, \sigma^2 I_D)$, so that $p_\beta(x|z) \sim N(g_\beta(z), \sigma^2 I_D)$. $g_\beta(z)$ can be a transformer, bert, gpt, bart for language learning.

For text modeling, $p_\beta(x|z)$ is an autoregressive model as shown in equation 10:

$$p_\beta(x|z) = \prod_t^T p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{10}$$

Let $p_{data}(x)$ be the data distribution, learning of β of $p_\beta(x)$ can be based on $\min_\beta KL(p_{data}(x)|p_{\beta(x)})$.

Observe $\{x_i, i = 1, ..., N\} \sim p_{data}(x)$,

$$\min_\beta KL(p_{data}(x)|p_\beta(x)) \approx L(\beta) = \frac{1}{N}\sum_i^N \log p_\beta(x_i) \tag{11}$$

which is the Maximum Likelihood Estimation (MLE). We calculate the gradient of $\log p_\beta(x)$:

$$\begin{aligned} \nabla_\beta \log p_\beta(x) &= \frac{1}{p_\beta(x)} \nabla_\beta p_\beta(x) \\ &= \frac{1}{p_\beta(x)} \int \nabla_\beta p_\beta(x, z) dz \\ &= E_{p_\beta(z|x)}[\nabla_\beta \log p_\beta(x, z)] \end{aligned} \tag{12}$$

The marginal distribution of x is $p_\beta(x) = \int p_\beta(x, z) dz$. The inference of z based on posterior is $p_\beta(z|x) = p_\beta(x, z)/p_\beta(x)$.

With gradient descent, in each iteration, we have:

$$\beta_{t+1} = \beta_t + \delta_t \frac{1}{N} E_{p_{\beta_t}(z_i|x_i)}[\nabla \log p_\beta(x_i, z_i)|\beta = \beta_t] \tag{13}$$

where $E_{p_{\beta_t}(z_i|x_i)}[...]$ is the gradient calculated by equation 12, approximated by short-run MCMC inference dynamics with Langevin dynamics.

We approximate the intractable expectation E_p with MCMC, Langevin dynamics, a gradience-based MCMC. We draw samples from the EBM prior:

$$z_{k+1} = z_k - s\nabla_z \log p_\alpha(z_k) + \sqrt{2s}\epsilon_k, \\ k = 1, ..., K, z_0 \sim p_0(z), \epsilon_k \sim N(0, I_d) \tag{14}$$

The short-run Langevin dynamics is always initialized from the fixed initial distribution p_0, and only runs a fixed number of K steps, e.g., $K = 20$.

Similarly, we can also draw samples from the posterior distribution $p_\beta(z|x)$:

$$z_{k+1} = z_k - s\nabla_z \log p_\beta(z_k|x) + \sqrt{2s}\epsilon_k, \\ k = 1, ..., K, z_0 \sim p_0(z), \epsilon_k \sim N(0, I_d) \tag{15}$$

where s is the small Langevin step size, t indexes the time step of the Langevin dynamics, $\nabla_z \log p_\alpha(z_k)$ or $\nabla_z \log p_\beta(z_k|x)$ can be efficiently computed by back propagation.

IV. Experiments

In this section, we assess the performance of our model on various tasks. In our experiment, we focus on task-incremental learning. We conduct a comparison of our method with eleven different techniques. All the methods use Mistral 7B as the backbone pretrained large language model. In the continual learning scenario, we train the model sequentially on a series of distinct tasks, following a predetermined order. After each training phase, we assess the model's performance on all previously encountered tasks.

A. Experimental Setup

1) Tasks, Datasets, and Metrics: We collect datasets for five different tasks related to natural language processing mentioned in decaNLP [25], including question answering, semantic parsing, sentiment analysis, semantic role labeling, and goal-oriented dialogue. To compare our method with [26], we also conduct experiments on four text classification tasks: news classification, sentiment analysis, Wikipedia article classification, and question-and-answer categorization with five datasets, following the same procedure for producing equal-sized datasets. Due to limited computational resources, we did not train on all datasets. We use a corresponding evaluation metric for each task. Table I summarizes the tasks, datasets, and metrics. The scores for the metrics range between 0 and 100%.

2) Methods for Comparison: The paper discusses various approaches to tackling the problem of catastrophic forgetting in continual learning, where a model trained on a sequence of tasks tends to forget the previous tasks when trained on subsequent tasks. The approaches considered in the paper include fine-tuning, multi-task learning, replay methods and architecture-based methods LAMOL, RVAE-LAMOL [27], HMI-LAMOL [28], PMR [29], regularization-based methods such as Online EWC [30] and MAS [14], Gradient Episodic Memory (GEM) [11], Improved Memory-Based Parameter Adaptation (MBPA++) [26], IDBR [15], and other methods like ProgPrompt [31]:

(1) **LSEBMCL**: Uses top-k sampling with $k = 1$. LSEBMCL$^{\gamma}_{GEN}$ denotes a sampling ratio γ, applying the same GEN token across tasks. (2) **LAMOL**: Employs $k = 20$ for top-k sampling and $\lambda = 0.25$ for LM loss weighting. (3) **RVAE-LAMOL**: Enhances LAMOL with a residual variational autoencoder. (4) **HMI-LAMOL**: Adds hippocampal memory indexing to improve generative replay via compressed features. (5) **PMR**: Stores minimal samples for efficient continual learning. (6) **Fine-tuning**: Sequentially trains tasks without task interaction. (7) **Multitask learning**: Trains all tasks simultaneously, serving as a continual learning upper bound. (8) **Regularization methods**: Includes Online EWC and MAS for mitigating forgetting. (9) **GEM**: Randomly samples 5% of prior task data for gradient calculation. (10) **MBPA++**: Combines sparse experience replay with local adaptation. (11) **IDBR**: Uses disentanglement-based regularization for text classification. (12) **ProgPrompt**: Prevents catastrophic forgetting without data replay or extensive task-specific parameters.

B. Experimental Results

1) SST, QA-SRL, and WOZ Tasks: To gain a preliminary understanding of the effectiveness of the different methods and the impact of the task order, we conducted an experiment on three small datasets: SST, QA-SRL, and WOZ. We trained all methods except for the multitasked method on six different orders of tasks. We evaluated the model's final score after training on each order, and the results are presented in Table II. Based on the results, we made several observations. We observed several things as follows: (1) Fine-tuned, EWC, MAS, GEM, LAMOL, and RVAE-LAMOL had worse performance than LSEBMCL even with $\gamma = 0$ and significantly worse than LSEBMCL with $\gamma > 0$. (2) LSEBMCL$^{0.2}_{GEN}$ achieves the best performance, even approximating the multitasked upper bound with 2.6%, implying little forgetting during continual learning. (3) Task order does influence performance with LSEBMCL. (4) When using LSEBMCL, the performance of old tasks remained almost the same throughout the training process. Increasing the sampling ratio γ improved the performance, particularly when increased from 0 to 0.05. (5) A better continual method had a smaller standard deviation, indicating it was less affected by the task order. LSEBMCL even achieves the lowest standard deviation among all the baselines, indicating its robustness to task order variations.

2) Five DecaNLP Tasks: In this sequential training experiment, five tasks were tackled in order of decreasing size, commencing with the largest task (SQuAD 2.0) and concluding with the smallest (WOZ). This task sequence was determined by the constraints of limited computing resources. Notably, LSEBMCL exhibited superior performance across all tasks, outperforming other methods by a significant margin and even approximates the multitasked upper bound with 0.9%, as detailed in Table III. Moreover, the effectiveness of LSEBMCL demonstrated further enhancement with an increase in the sampling ratio γ. The experiment's results underscore LSEBMCL's remarkable efficacy and suitability for diverse tasks, confirming its robust performance.

3) Text Classification Tasks: We compare our proposed method, LSEBMCL, against the state-of-the-art MBPA++,

TABLE I

SUMMARY OF TASKS, DATASETS, DATASET SIZES, AND THEIR CORRESPONDING METRICS. AS THIS WORK USES NO DEVELOPMENT SET, ONLY THE TRAINING AND TEST DATASETS ARE SHOWN. NF1 IS THE NORMALIZED VERSION OF THE F1 SCORE; EM REPRESENTS AN EXACT MATCH BETWEEN TEXTS: FOR TEXT CLASSIFICATION, THIS AMOUNTS TO ACCURACY; FOR WOZ, IT IS EQUIVALENT TO DFEM (TURN-BASED DIALOGUE STATE EXACT MATCH); FOR WIKISQL, IT IS EQUIVALENT TO LFEM (EXACT MATCH OF LOGICAL FORMS).

Task	Dataset	# Train	# Test	Metric
Question answering	SQuAD 2.0	130319	11873	nF1
Semantic parsing	WikiSQL	56355	15878	lfEM
Sentiment analysis	SST	6920	1821	EM
Semantic role labeling	QA-SRL	6414	2201	nF1
Goal-oriented dialogue	WOZ	2536	1646	dsEM
Text classification	AGNews Amazon DBPedia Yahoo Yelp	115000	7600	EM

TABLE II

SUMMARY OF AVERAGED METRIC SCORES FOR DIFFERENT METHODS UNDER PERMUTED TASK ORDERS USING MODELS AT THE LAST EPOCH OF THE LAST TASK. THE AVERAGE AND STD COLUMNS RESPECTIVELY ARE THE AVERAGE AND STANDARD DEVIATION OF THE AVERAGED SCORES FOR EACH ROW OF THE METHODS. MULTITASKED LEARNING AS AN UPPER BOUND IS SHOWN AT THE BOTTOM.

Model	SST SRL WOZ	SST WOZ SRL	SRL SST WOZ	SRL WOZ SST	WOZ SST SRL	WOZ SRL SST	Average	Std
Fine-tuned	52.0	25.5	64.6	33.2	34.6	34.6	40.8	14.6
EWC	51.0	50.0	66.4	37.5	44.8	40.7	48.4	10.2
MAS	37.3	46.2	57.3	32.3	50.3	32.2	42.6	10.3
GEM	52.1	31.5	64.0	32.9	45.3	36.7	43.8	12.6
LAMOL $^{0}_{GEN}$	47.1	38.4	57.8	39.4	45.5	46.6	45.8	7.0
LAMOL $^{0.05}_{GEN}$	81.5	79.5	74.8	73.7	70.2	75.8	75.9	4.1
LAMOL $^{0.2}_{GEN}$	80.8	81.1	81.2	80.3	78.9	81.9	80.7	1.0
RVAE-LAMOL $^{0.05}_{GEN}$	80.7	79.6	79.9	80.2	79.6	78.0	79.7	0.9
RVAE-LAMOL $^{0.2}_{GEN}$	81.5	82.3	81.5	82.0	81.1	82.8	81.9	0.6
LSEBMCL $^{0}_{GEN}$	66.4	57.7	77.7	66.8	64.6	65.6	66.5	6.4
LSEBMCL $^{0.05}_{GEN}$	82.8	81.8	**84.4**	81.9	82.7	80.5	82.4	1.3
LSEBMCL $^{0.2}_{GEN}$	**83.1**	**82.5**	82.7	**82.4**	**83.7**	**83.2**	**82.9**	**0.5**
Multitasked	87.0							

TABLE III

SUMMARY OF THE AVERAGED SCORE ON FIVE TASKS. THE SEQUENCE ORDER IS SQUAD 2.0, WIKISQL, SST, QA-SRL, AND WOZ. THE SCORES ARE REPORTED AS THE AVERAGED SCORE OVER ALL TASKS OF THE MODELS AFTER TRAINING ON EVERY TASK. THE RIGHTMOST COLUMN MULTITASKED IS THE UPPER BOUND FOR COMPARISON. THE BEST PERFORMANCE IS IN BOLDFACE.

Fine-tuned	MAS	LAMOL $^{0.05}_{GEN}$	LAMOL $^{0.2}_{GEN}$	HMI-LAMOL$^{0.05}_{GEN}$	HMI-LAMOL$^{0.2}_{GEN}$	LSEBMCL $^{0.05}_{GEN}$	LSEBMCL $^{0.2}_{GEN}$	Multitasked
52.6	51.2	70.3	74.1	76.0	76.9	**76.5**	**77.3**	78.2

TABLE IV

SUMMARY OF RESULTS ON TEXT CLASSIFICATION TASKS USING AVERAGED EM SCORE (EQUIVALENT TO AVERAGED ACCURACY IN [26]) OF MODELS AT LAST EPOCH OF LAST TASK. THE FOUR ORDERS MIRROR THOSE IN [26]. FOR MBPA++, MBPA++ (OUR IMPL.), LAMOL $^{0.2}_{TASK}$, PMR, IDBR, PROGPROMPT, HMI-LAMOL, AND LSEBMCL$^{0.05}_{GEN}$, THE RESULTS ARE AVERAGED OVER TWO RUNS.

Order	MBPA++	MBPA++ (our impl.)	LAMOL $^{0.2}_{TASK}$	PMR	IDBR	ProgPrompt	HMI-LAMOL	LSEBMCL $^{0.05}_{GEN}$
i	70.8	75.3	78.6	73.5	77.0	78.9	77.8	**80.2**
ii	70.9	76.0	78.3	74.3	77.2	78.4	78.7	**80.0**
iii	70.2	73.9	78.0	72.0	78.1	79.0	79.7	**80.2**
iv	70.7	76.7	76.9	70.2	77.8	78.0	78.4	**80.3**
Average	70.7	75.5	77.9	72.5	77.5	79.0	78.6	**80.2**

LAMOL, PMR, IDBR, ProgPrompt, and HMI-LAMOL. The results are shown in Table IV. LSEBMCL$^{0.05}_{GEN}$ outperformed LAMOL$^{0.2}_{TASK}$, our implementation of MBPA++, PMR, IDBR, ProgPrompt, and HMI-LAMOL even with sampling ratio 0.05 and GEN token. This indicates that the improvements made in LSEBMCL were significant and that it is a strong method for mitigating catastrophic forgetting with less sampling data.

V. Conclusion

In this study, we introduce an innovative approach known as LSEBMCL that integrates an EBM layer into the continual learning framework for NLP tasks. In addition to its promising applications in NLP tasks, it holds potential implications for computer vision tasks. Leveraging the expressive power of the EBM prior in text modeling, we construct a latent space conducive to interpretable generation and text classification. To achieve this, we devise a novel prior distribution that integrates continuous latent variables for generation and discrete latent variables for inducing structural elements. Furthermore, we utilize the EBM to generate samples from previous tasks when training the model on new tasks. The experiments demonstrate the superior performance of our proposed approach.

Acknowledgement

The research reported herein was supported in part by NSF awards DMS-1737978, DGE-2039542, OAC-1828467, OAC-1931541, and DGE-1906630, ONR awards N00014-17-1-2995 and N00014-20-1-2738, Army Research Office Contract No. W911NF2110032 and IBM faculty award (Research).

References

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[2] B. Pang, T. Han, E. Nijkamp, S.-C. Zhu, and Y. N. Wu, "Learning latent space energy-based prior model," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 994–22 008, 2020.

[3] S. Li, Y. Du, G. van de Ven, and I. Mordatch, "Energy-based models for continual learning," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 1–22.

[4] Y. Zhao, Y. Zheng, Z. Tian, C. Gao, B. Yu, H. Yu, Y. Li, J. Sun, and N. L. Zhang, "Prompt conditioned vae: Enhancing generative replay for lifelong learning in task-oriented dialogue," *arXiv preprint arXiv:2210.07783*, 2022.

[5] V. Varshney, M. Patidar, R. Kumar, L. Vig, and G. Shroff, "Prompt augmented generative replay via supervised contrastive learning for lifelong intent detection," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1113–1127.

[6] D. Li, Z. Chen, E. Cho, J. Hao, X. Liu, F. Xing, C. Guo, and Y. Liu, "Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 5441–5454.

[7] G. Li, Y. Zhai, Q. Chen, X. Gao, J. Zhang, and Y. Zhang, "Continual few-shot intent detection," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 333–343.

[8] Q. Zhu, B. Li, F. Mi, X. Zhu, and M. Huang, "Continual prompt tuning for dialog state tracking," *arXiv preprint arXiv:2203.06654*, 2022.

[9] C. Qin and S. Joty, "LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=HCRVf71PMF

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[11] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, pp. 6467–6476, 2017.

[12] F.-K. Sun, C.-H. Ho, and H.-Y. Lee, "Lamol: Language modeling for lifelong language learning," *arXiv preprint arXiv:1909.03329*, 2019.

[13] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[14] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[15] Y. Huang, Y. Zhang, J. Chen, X. Wang, and D. Yang, "Continual learning for text classification with information disentanglement based regularization," *arXiv preprint arXiv:2104.05489*, 2021.

[16] X. Li, Z. Wang, D. Li, L. Khan, and B. Thuraisingham, "Lpc: A logits and parameter calibration framework for continual learning," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 7142–7155.

[17] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[18] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4548–4557.

[19] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

[20] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, "A theory of generative convnet," in *International Conference on Machine Learning*. PMLR, 2016, pp. 2635–2644.

[21] E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu, "Learning non-convergent non-persistent short-run mcmc toward energy-based model," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] T. He, B. McCann, C. Xiong, and E. Hosseini-Asl, "Joint energy-based model training for better calibrated natural language understanding models," *arXiv preprint arXiv:2101.06829*, 2021.

[23] S. Yoon, Y.-U. Jin, Y.-K. Noh, and F. C. Park, "Energy-based models for anomaly detection: A manifold diffusion recovery approach," *arXiv preprint arXiv:2310.18677*, 2023.

[24] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[25] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[26] C. de Masson D'Autume, S. Ruder, L. Kong, and D. Yogatama, "Episodic memory in lifelong language learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] H. Wang, R. Fu, X. Zhang, and J. Zhou, "Rvae-lamol: Residual variational autoencoder to enhance lifelong language learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–9.

[28] A. Maekawa, H. Kamigaito, K. Funakoshi, and M. Okumura, "Generative replay inspired by hippocampal memory indexing for continual language learning," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 930–942.

[29] S. Ho, M. Liu, L. Du, L. Gao, and Y. Xiang, "Prototype-guided memory replay for continual learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[30] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *International conference on machine learning*. PMLR, 2018, pp. 4528–4537.

[31] A. Razdaibiedina, Y. Mao, R. Hou, M. Khabsa, M. Lewis, and A. Almahairi, "Progressive prompts: Continual learning for language models," *arXiv preprint arXiv:2301.12314*, 2023.