Iterative Self-Synthesized Rehearsal and Elastic-Variational Continual Learning for Catastrophic Forgetting

Pranav Sharma , Shreya Shetye, Vibhor Tyagi , Yash Samir Kakde
University of Illinois, Urbana-Champaign
pranav24@illinois.edu , sshety3@illinois.edu , vtyag@illinois.edu, kakde2@illinois.edu

ABSTRACT

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) with exceptional performance in tasks like language understanding, text generation, and question answering. However, their use in sequential learning is limited by catastrophic forgetting where previously learned knowledge is lost while adapting to new tasks. Existing methods like Elastic Weight Consolidation (EWC) mitigate this by preserving key model parameters but overlook uncertainty in parameter importance. Similarly, Self-Synthesized Rehearsal (SSR), which generates synthetic rehearsal data, often suffers from low-quality outputs, undermining performance.

We propose a novel framework integrating Elastic-Variational Continual Learning (EVCL) with an iterative feedback-enhanced SSR mechanism. While current EVCL approaches focus on neural networks in computer vision, we extend this concept to NLP, creating a pioneering system that iteratively refines SSR through feedback between a base LLM and an evaluator. Using quantized LLaMA 3 8B and 3.2 3B models, along with in-context learning, our method ensures high-quality synthetic datasets for continual learning. Initial experiments on low-resource settings show improved task retention compared to standard LoRA-based Peft fine-tuning and vanilla SSR, showcasing its potential for real-world continual learning applications.

Link to Github Repository

1 Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) by achieving state-of-the-art results in tasks such as summarization, translation, and conversational AI. Their ability to understand context, generate coherent text, and solve complex problems has established them as invaluable tools across diverse applications, including automated customer support and creative content generation. As these models grow in scale and complexity, they extend the horizons of NLP, paving pathways for advanced human-computer interactions and sophisticated decision-making systems.

Despite these advancements, LLMs face significant challenges in continual learning, particularly the phenomenon of **catastrophic forgetting**, where previously acquired knowledge is lost while learning new tasks. This issue is critical in dynamic real-world environments, where integrating new knowledge without discarding the old is essential. For example, an AI-driven healthcare system must incorporate new diagnostic procedures without forgetting established ones to maintain consistent and reliable medical assessments.

Catastrophic forgetting occurs when updates for new tasks overwrite parameters crucial for earlier tasks, leading to degraded performance on previously learned tasks. Traditional fine-tuning methods often fail to balance learning new information (plasticity) with retaining old knowledge (stability). To address this, strategies such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) and Self-Synthesized Rehearsal (SSR) (Huang et al., 2024) have been proposed.

EWC mitigates forgetting by adding constraints that protect important parameters related to past tasks during updates. However, it assumes parameter importance remains constant, which may not hold in dynamically changing environments. SSR generates synthetic rehearsal data to preserve earlier competencies, but its effectiveness depends heavily on the quality of the generated data. Low-quality synthetic instances can mislead the model and impair generalization.

Self-Synthesized Rehearsal (SSR)
SSR addresses catastrophic forgetting by generating synthetic data using the LLM itself, eliminating the need to store real training data from previous tasks. Diverse synthetic instances are selected for rehearsal, preserving the model's capabilities while learning new tasks. However, ensuring the quality of synthetic data remains a challenge. To address this, we developed an iterative refinement process using an Evaluator LLM (Prometheus), which provides feedback to improve data quality. This feedback mechanism enhances SSR's ability to balance task retention, generalization, and resource efficiency, outperforming conventional approaches and improving robustness in real-world continual learning scenarios.

Variational Continual Learning (VCL)
VCL (Nguyen et al., 2017) employs a Bayesian framework to approximate the posterior distribution of model parameters for sequential tasks. Variational inference updates the posterior recursively by maximizing the

Evidence Lower Bound (ELBO) and minimizing the Kullback-Leibler (KL) divergence (Shlens, 2014) between the current and previous task posteriors. This allows VCL to retain past knowledge without revisiting prior data, making it a scalable solution. However, VCL's reliance on coresets and susceptibility to accumulated errors limit its scalability and flexibility.

Elastic Weight Consolidation (EWC)
EWC uses the Fisher Information Matrix (FIM) to identify parameters critical to prior tasks. By penalizing updates to these parameters, EWC helps retain essential knowledge while adapting to new tasks. However, EWC's reliance on a local Laplace approximation can underestimate parameter importance and fail to capture complex dependencies, reducing its effectiveness for diverse sequential tasks.

Combining VCL and EWC in EVCL
Elastic Variational Continual Learning with Weight Consolidation (EVCL) integrates EWC's regularization within the VCL framework. EWC leverages the Fisher Information Matrix to identify crucial parameters, safeguarding them during updates. Combined with VCL's posterior approximation, EVCL balances learning new information and retaining knowledge from prior tasks. The core EVCL loss function is defined as:

$$\mathcal{L}^t_{\text{EVCL}}(q_t(\theta)) = \mathcal{L}^t_{\text{VCL}}(q_t(\theta)) + \sum_i \frac{\lambda}{2} F_{t-1,i}\Big[(\mu_{t,i} - \mu_{t-1,i})^2 + (\sigma^2_{t,i} - \sigma^2_{t-1,i})^2\Big] \tag{1}$$

Here, $\mathcal{L}^t_{\text{VCL}}(q_t(\theta))$ is the VCL objective for task t, $\mu_{t,i}$ and $\sigma^2_{t,i}$ are the mean and variance of the parameter posterior, and $F_{t-1,i}$ is the Fisher Information Matrix from the previous task, measuring parameter importance. The term λ controls the strength of regularization. This formulation reduces the need for coresets by using the Fisher Information Matrix to identify essential parameters, guiding the model to preserve knowledge while adapting to new tasks.

In this research, we make the following contributions:

- We improve the data quality issues in SSR by incorporating an Evaluator LLM (Prometheus) for feedback-driven iterative refinement of synthetic data.
- We extend the EVCL framework, previously applied to neural networks in computer vision, to LoRA-based LLMs for NLP tasks.
- We develop a hybrid framework combining high-quality synthetic data with EVCL for mitigating catastrophic forgetting in LLMs.

We observed that our proposed methodology outperforms LoRA-based fine-tuning and SSR-only fine-tuning in mitigating catastrophic forgetting, with more pronounced effects on larger models (LLaMA 3 (8B)) (Dubey et al., 2024) compared to smaller models (Llama 3.2 (3B)) in low-resource environments.

2 Related Work

Continual Learning (CL) is crucial for enabling machine learning models to learn tasks sequentially while retaining previously acquired knowledge. However, catastrophic forgetting - the tendency of a model to overwrite prior knowledge when learning new tasks, remains a significant challenge in CL. Despite substantial progress in mitigating forgetting, existing methods often struggle to balance task retention, overfitting reduction, and data privacy.

SSR has emerged as a promising technique to address these challenges. Huang et al. (2024) introduced SSR as a mechanism for generating synthetic data within the model itself, bypassing the need for stored datasets. By mitigating privacy concerns and storage constraints, SSR enhances data efficiency while enabling models to maintain task-specific knowledge. However, synthetic data quality remains a key bottleneck, as low-quality synthetic samples can hinder generalization and exacerbate overfitting. The proposed integration of an iterative feedback loop to refine synthetic data quality builds on this foundation, ensuring better task retention and enhanced model performance.

Regularization-based methods, such as EWC, have also played a pivotal role in mitigating catastrophic forgetting. EWC, introduced by Kirkpatrick et al. (2016), constrains parameter updates by penalizing changes to weights deemed critical for previously learned tasks. While EWC ensures task retention, it does not address challenges related to data quality and synthesis, limiting its utility in hybrid frameworks.

To overcome the limitations of both SSR and EWC, variational inference-based methods such as EVCL have been developed. Batra and Clark (2024) demonstrated that EVCL effectively balances learning stability and plasticity by modeling parameter uncertainty using Evidence Lower Bound Optimization (ELBO). By dynamically adjusting model parameters based on task-specific uncertainty, EVCL addresses catastrophic forgetting while maintaining adaptability. The integration of EVCL with SSR in our proposed framework offers a novel approach to leveraging parameter uncertainty alongside high-quality synthetic data for enhanced continual learning.

Empirical studies further highlight the importance of hybrid approaches in CL. Luo et al. (2023) (Luo et al., 2023) analyzed catastrophic forgetting during continual fine-tuning of large language models, revealing trade-offs between generalization and task-specific performance. Similarly, Li et al. (2024) (Li et al., 2024) examined parameter-efficient tuning strategies and emphasized the need for techniques that reduce overfitting while retaining task knowledge. Replay-based methods, such as those explored by Rostami et al. (2019) (Rostami et al., 2019), have shown that combining re-

play with other strategies improves memory retention. However, SSR offers a unique advantage over traditional replay by generating synthetic data internally, eliminating the need for explicit data storage.

The proposed framework addresses catastrophic forgetting in LLMs by integrating Elastic-Variational Continual Learning (ELBO-VCL) with an enhanced Self-Synthesized Rehearsal (SSR) mechanism. This methodology is designed to generate high-quality synthetic rehearsal data, effectively model parameter uncertainty, and mitigate forgetting during sequential learning.

3 Methodology: Iterative SSR with EVCL

In this work, we extend the Elastic Variational Continual Learning with Weight Consolidation (EVCL) framework, originally applied to multi-layer perceptrons (MLPs) in computer vision (CV), to large language models (LLMs) using Low-Rank Adaptation (LoRA) (Hu et al., 2021) weights. Additionally, we integrate a novel Synthetic Data Rehearsal (SSR) mechanism, where synthetic data is generated, refined, and evaluated to enhance continual learning. This allows for efficient task rehearsal and robust retention of knowledge from previous tasks. Below, we describe our methodology as shown in 1, integrating SSR into the EVCL process for LLMs.

3.1 Initialization with Variational Continual Learning (VCL)

As no prior tasks exist initially, the first step involves initializing the LoRA weights variationally using Variational Continual Learning (VCL). The pretrained LLM provides the initial parameter means, and uncertainty is introduced to model variance, which represents potential variability in the weights. This Bayesian framework allows for quantifying confidence in the parameters. The model is finetuned on the first task using variational inference, updating the posterior mean and variance for the LoRA weights. As no regularization is needed for this initial task, the focus is on establishing a posterior distribution for the parameters. Once finetuning is complete, the model state, including posterior means and variances, is saved using the Pyro (Bingham et al., 2019) framework. Additionally, synthetic data generation for task rehearsal begins at this stage, leveraging SSR to create a high-quality rehearsal dataset for task 1.

3.2 SSR Implementation

Synthetic data generation is a crucial component of our framework, employing both the LLaMA 3 8B (Dubey et al., 2024) and LLaMA 3.2 3B (Bib, 2024) models to generate, refine, and evaluate rehearsal data. We utilize these two distinct configurations of Large Language Models — an 8B model and a 3.2B model — in separate, sequential pipelines to perform comprehensive comparative analyses. Each model independently undergoes the same process, initially used for synthetic data generation and subsequently for iterative refinement tasks within its designated pipeline.

The 8B model, quantized to 8-bit precision, is utilized first. This model strikes a balance between computational efficiency and performance, making it ideal for extensive testing across various NLP tasks. It is tasked with generating and refining synthetic data, essential for mitigating catastrophic forgetting by enhancing the model's ability to retain prior knowledge while integrating new information.

Following the completion of tasks with the 8B model, the same procedures are replicated using the 3.2B model. Despite its smaller size, this model is employed to assess how a more compact architecture performs under identical conditions. The tasks involving this model include generating and refining data in a manner akin to that of the 8B model, enabling a direct comparison of the impacts of model size and complexity on task retention, generalization, and resource efficiency.

By analyzing the outcomes from both models separately, our framework elucidates the differential capabilities and efficiencies of these models in continual learning scenarios. This structured comparative approach not only bolsters the robustness of our findings but also deepens our understanding of the trade-offs involved when scaling model architectures for specific NLP tasks.

The SSR process initiates with the generation of synthetic data through in-context learning techniques. Both models independently produce 2000 synthetic data instances, driven by prompts designed to reflect patterns and scenarios pertinent to the target tasks. These prompts are crafted to encourage the models to create instances that simulate a variety of task-specific scenarios, enhancing the relevance and applicability of the data across different domains.

To ensure quality and diversity, a semantic clustering technique, such as K-Means, is applied to group similar instances based on semantic similarity metrics. From each cluster, a representative subset of 200 instances is selected to maintain task diversity and minimize redundancy.

Refinement occurs through an iterative feedback loop involving the Prometheus evaluation model. Each sampled instance is evaluated on a scale from 1 to 5, assessing its alignment with task requirements, coherence, and relevance. Instances scoring below 4 are resubmitted for refinement, with specific feedback provided to guide improvements in areas such as clarity or adherence to task-specific nuances. This process is repeated until an instance achieves a score of 4 or higher or undergoes two iterations, ensuring that only high-quality rehearsal data is incorporated. This feedback-driven refinement process distinguishes our SSR mech-

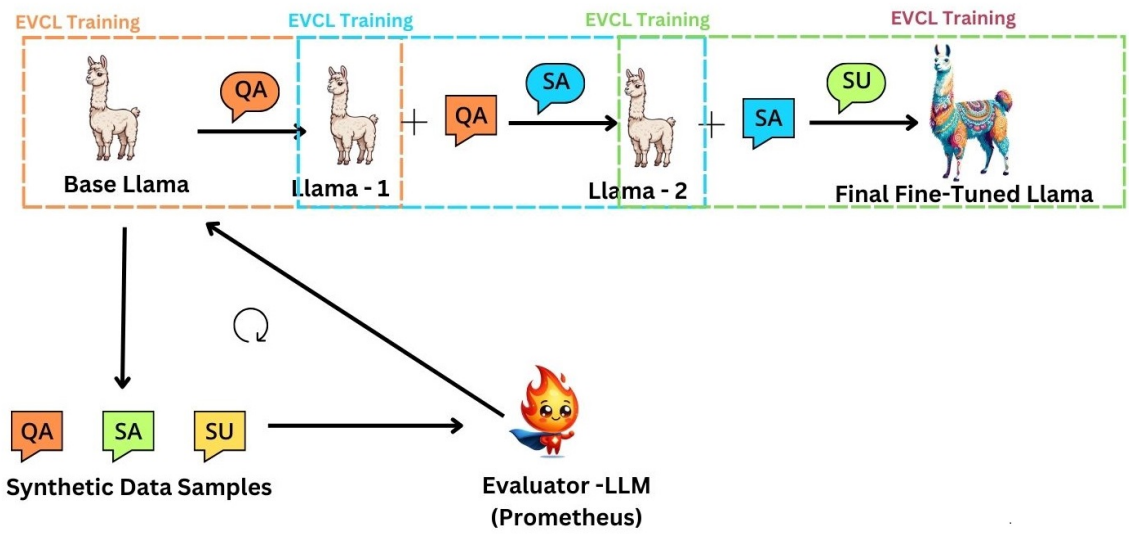

Figure 1: Our framework for Continual Learning with Iterative SSR and EVCL. The framework sequentially fine-tunes a base LLM (Llama) across multiple tasks Question Answering (QA), Sentiment Analysis (SA), and Summarization (SU) using (EVCL). Synthetic rehearsal data is generated during training and refined iteratively through a feedback loop with an evaluator LLM (Prometheus) ensuring high-quality synthetic data for each task, enabling effective knowledge retention and mitigating catastrophic forgetting across tasks.

anism from traditional approaches, where data quality is static and unverified.

3.3 Fisher Information Matrix (FIM) Calculation

After learning the first task, the Fisher Information Matrix (FIM) is computed to identify the most critical LoRA parameters for task 1. The FIM quantifies the sensitivity of the model's loss to parameter changes, enabling the identification of weights most critical for retaining task-specific knowledge. Alongside the FIM, posterior means of the LoRA parameters are saved to serve as priors for subsequent tasks. The high-quality rehearsal data generated by SSR for task 1 is also finalized at this stage. This synthetic dataset is preserved and updated dynamically in subsequent tasks, providing a consistent yet evolving rehearsal mechanism to mitigate catastrophic forgetting.

3.4 Continual Learning with EVCL and SSR for Subsequent Tasks

For each new task t, the model parameters are initialized with the posterior means and variances from task $t-1$, and the FIM is loaded to preserve critical parameters. The SSR mechanism is employed to update and refine the rehearsal dataset dynamically for all previous tasks $1, \ldots, t-1$, ensuring the synthetic data remains relevant to the evolving model.

The updated rehearsal data from SSR is integrated into the learning process by incorporating synthetic instances alongside the real task data during finetuning. This allows the model to rehearse knowledge from prior tasks while adapting to the new task. The EVCL loss function is applied, balancing the VCL objective and EWC regularization:

$$\mathcal{L}^t_{\text{EVCL}}(q_t(\theta)) = \mathcal{L}^t_{\text{VCL}}(q_t(\theta)) + \sum_i \frac{\lambda}{2} F_{t-1,i} \Big[(\mu_{t,i} - \mu_{t-1,i})^2 + (\sigma^2_{t,i} - \sigma^2_{t-1,i})^2 \Big] \tag{2}$$

Here, $\mathcal{L}^t_{\text{VCL}}(q_t(\theta))$ represents the variational objective for task t, $\mu_{t,i}$ and $\sigma^2_{t,i}$ denote the mean and variance of the parameter posterior, and $F_{t-1,i}$ reflects the Fisher Information Matrix for task $t-1$.

SSR-generated rehearsal data ensures that the model remains aware of prior tasks by providing task-relevant context-question pairs, while the FIM ensures critical parameters are protected. This dual mechanism as shown in 1 allows the model to balance plasticity (learning new tasks) and stability (retaining prior knowledge).

3.5 Evaluation and Iteration

After finetuning on each task t, the model is evaluated across all tasks $1, \ldots, t$, assessing its performance on both the newly learned task and previous tasks. This step ensures the effectiveness of SSR in retaining knowledge and the success of FIM-based regularization in preventing catastrophic forgetting. SSR rehearsal datasets are further updated after task evaluation, incorporating any refinements needed to improve task diversity and alignment with evolving model requirements.

3.6 Adapting EVCL from CV to NLP

Unlike the original EVCL framework, which was applied to image data in CV, this work adapts the framework for NLP tasks using LoRA weights in LLMs. While CV tasks typically involve static and low-dimensional data, NLP tasks require handling high-dimensional, structured text data that evolves over sequences and domains. This introduces additional challenges in maintaining coherence and task relevance across multiple tasks.

SSR addresses these challenges by dynamically generating, refining, and evaluating synthetic rehearsal data for NLP tasks. By leveraging LLaMA 3 models, the SSR mechanism ensures that rehearsal data captures task-specific nuances while maintaining quality and diversity. This rehearsal mechanism, combined with the Bayesian framework of EVCL, enables LLMs to adapt to new tasks while preserving knowledge from previous ones.

By focusing on LoRA weights, this approach also ensures computational efficiency, making it feasible to apply EVCL and SSR to large-scale LLMs without incurring significant memory or compute overhead.

Algorithm 1 Iterative SSR with EVCL for Continual Learning

Initialize LoRA weights using Variational Continual Learning (VCL):
$(\text{Mean}, \text{Variance}) \leftarrow \text{InitializeVariational(Model)}$
Compute posterior distribution after Task 1:
$\text{Posterior} \leftarrow \text{VariationalInference(Task\_1\_Data, Mean, Variance)}$
Generate synthetic rehearsal data using SSR:
$\text{RehearsalData} \leftarrow \text{GenerateSyntheticData(Task\_1\_Data)}$
Save posterior means and variances:
SavePosterior(Posterior)
for Task $t = 2$ to T **do**
    Load Posterior and Fisher Information Matrix (FIM) from Task $t-1$:
    $(\text{Posterior}, \text{FIM}) \leftarrow \text{LoadTaskData(t-1)}$
    Refine synthetic rehearsal data:
    $\text{RefinedData} \leftarrow \text{RefineSyntheticData(RehearsalData, Evaluator)}$
    for Batch $\in$ Task_t_Data + RefinedData **do**
        Update LoRA weights using EVCL loss:
        $\text{Loss} \leftarrow \text{ComputeEVCLLoss(Batch, Posterior, FIM, } \lambda)$
        Fine-tune weights via gradient descent
    end for
    Save updated posterior means and variances:
    SavePosterior(Task_t)
end for
Evaluate model on all tasks $\{1, ..., T\}$
return Updated Model, Posterior

4 Experiments

Our experimental evaluation is designed to assess the efficacy of our proposed synthetic data generation and refinement methodology using two configurations of LLaMA-based models. These models were selected to investigate the impact of model scale and computational efficiency on task performance in continual learning scenarios. The details of the models used are as follows:

- **LLaMA 3 (8B)**: This model is quantized to 8-bit precision, which optimizes it for efficient training and evaluation while retaining robust performance capabilities. It is used in one of our experimental pipelines to generate and refine synthetic data, serving as our primary model for assessing the impact of our methods on a larger scale.
- **LLaMA 3.2 (3B)**: A more compact and computationally less demanding variant, this model is specifically utilized for iterative refinement tasks in a separate pipeline. Its smaller size allows us to explore how well refined and efficient architectures manage continual learning tasks compared to their larger counterparts.

Each model undergoes a distinct testing pipeline where they are independently tasked with generating, refining, and evaluating synthetic data. This setup allows us to perform a comparative analysis across the two models, focusing on their ability to retain knowledge from previously learned tasks while effectively integrating new information. The goal is to identify which model dimensions and configurations yield the best balance between computational efficiency and task performance in real-world NLP applications.

4.1 Dataset and Tasks

Our experiments are conducted on the SuperNI (Wang et al., 2022) dataset, a comprehensive benchmark for instruction tuning that includes tasks from diverse domains. To simulate a typical continual learning scenario, we select a subset of three tasks from SuperNI: **QA** (Question Answering) $\rightarrow$ **SA** (Sentiment Analysis) $\rightarrow$ **Sum** (Summarization).

This order mirrors real-world task progression and simplifies empirical comparisons. For each task, we sample 2,000 training instances and 500 evaluation instances to ensure sufficient representation while maintaining computational efficiency.

4.2 Baselines

We compare our approach against the following baselines:

- **Non-Rehearsal**: A naive baseline where the model is fine-tuned solely on the current task data without leveraging prior task knowledge.

- **SSR**: Utilizes synthetic data from previous tasks alongside the training data of the current task.
- **Proposed Method (SSR-EVCL)**: Combines our refined SSR methodology with Elastic-Variational Continual Learning (EVCL), introducing a novel approach to address catastrophic forgetting in NLP.

4.3 Evaluation Metrics

Given the diversity and open-ended nature of SuperNI tasks, we adopt **ROUGE-L** (Lin, 2004) as the primary metric, which aligns well with human evaluation. Additionally, we include the following metrics for deeper analysis:

- **Average ROUGE-L (AR)**: Quantifies the final average performance across all tasks after the final training stage T:

$$AR = \frac{1}{T}\sum_{i=1}^{T} a(T)_i.$$

- **Backward Transfer (BWT)**: Measures the impact of learning new tasks on previous tasks, comparing the final performance $a(T)_i$ to the online performance $a(i)_i$:

$$BWT = \frac{1}{T-1}\sum_{i=1}^{T-1} \big(a(T)_i - a(i)_i\big).$$

  A negative BWT indicates forgetting of previously acquired knowledge.

4.4 Implementation Details

In our study, we employ the LoRA (Low-Rank Adaptation) technique (Hu et al., 2021) to efficiently fine-tune the self-attention mechanism of our models by selectively updating only the query and value projection matrices. This method enables significant computational savings while maintaining high performance, making it ideal for handling large-scale models like those used in our experiments.

For the **LLaMA 3 (8B)** model, we configure the LoRA adaptation with a rank of 8 and a dropout rate of 0.1 to mitigate overfitting. The initial learning rate is set at 2×10^{-4}, optimized to balance rapid convergence with the stability required for effective learning across multiple tasks. This model is operated in 8-bit precision to enhance computational efficiency. We further utilize the Fisher Information Matrix and Elastic Weight Consolidation (EWC), with parameters set at ewc_gamma $= 1.0$ and ewc_lambda $= 50$, to preserve critical parameters from prior learning while accommodating new tasks. Training configurations include a duration of 100 epochs per task, using a batch size of 8 to ensure detailed attention to learning nuances without overloading computational resources.

Similarly, the **LLaMA 3.2 (3B)** model is fine-tuned with LoRA parameters adjusted to a lower rank of 4, reflecting its smaller scale and reduced computational requirements. The same dropout rate and learning rate are applied as in the 8B model to maintain consistency in training behavior. Fisher Information and EWC settings remain unchanged to standardize the approach to mitigating catastrophic forgetting across different model scales. This model undergoes a shorter training cycle of 11 epochs per task, which is adequate given its architecture and the focused nature of the tasks it handles.

To ensure uniformity across experiments, we adopt a global batch size of 32 for all tasks, regardless of model configuration. Input lengths are capped at 1,024 tokens, and output lengths at 512 tokens for both models, optimizing the balance between input detail and manageable computation loads. These hyperparameters are meticulously chosen to align computational efficiency with the models' adaptability to new tasks and the retention of knowledge from previous tasks, thereby demonstrating the effectiveness of the EVCL framework for continual learning in LLMs.

4.5 Synthetic Data Generation and Clustering

To facilitate in-context learning (ICL), we employ 1% of the SuperNI training data as demonstration examples, from which we sample K = 2 demonstrations multiple times. This approach allows us to generate a diverse set of synthetic instances that enrich the learning process. For the organization and analysis of this synthetic data, we use K-means clustering with C = 20 clusters. This method ensures that the data is not only diverse but also representative of the various types of scenarios that the models might encounter, thereby maximizing the effectiveness of the synthetic data in training.

After an initial refinement and sampling, we employed the Prometheus model as a crucial component for the iterative refinement process of synthetic data. This model serves to evaluate the initial synthetic instances generated by the LLM, providing a rigorous assessment based on predefined criteria such as relevance, coherence, and alignment with task-specific requirements. Following this evaluation, the Prometheus model generates detailed feedback for each instance.

This feedback is then fed back into the latest version of the LLM, which uses the insights to refine the synthetic data further. This process iteratively enhances the quality of the synthetic instances, ensuring that each batch of data progressively aligns more closely with the real-world demands of the tasks at hand. The loop continues until the synthetic data meets the established quality standards or for a maximum of two iterations, optimizing both the effectiveness of the synthetic rehearsal and the efficiency of the learning process.

This feedback-driven refinement cycle, mediated by the Prometheus model, is integral to our methodology,

allowing for continuous improvement and adaptation of the synthetic data used in training the LLM. This mechanism not only improves the model's performance on current tasks but also enhances its generalization capabilities to new, unseen tasks, showcasing the dynamic adaptability of our approach.

Our experimental framework evaluates the efficacy of the proposed SSR-EVCL method, focusing on its impact on task retention, generalization, and resource efficiency. This comprehensive analysis allows us to assess how well our framework performs in comparison to state-of-the-art baselines, providing crucial insights into its potential utility in real-world applications.

5 Results

The evaluation of our methods across various NLP tasks and configurations provides key insights into their performance. Below, we discuss the findings shown in 1 and 2 in detail with references to the specific values observed in the results.

5.1 Self-Synthesized Rehearsal (SSR) vs. EVCL-SSR

SSR demonstrated competitive performance across tasks but was slightly outperformed by the combination of EVCL with SSR. For the 8B model 1, SSR achieved an average ROUGE-L score of 26.34 for the QA-SA-Summ configuration, while EVCL-SSR achieved 28.37, indicating a significant improvement of 2.03 points. A similar trend was observed in the 3B model 2, where SSR's QA-SA-Summ score was 18.95, while EVCL-SSR slightly outperformed with 18.59. The observed improvement for the 8B model is more pronounced, reflecting the effectiveness of EVCL-SSR in larger models.

The improved performance of EVCL-SSR can be attributed to its ability to model parameter uncertainty, enhancing stability during continual learning and mitigating catastrophic forgetting. These results indicate that while SSR is effective on its own, integrating EVCL provides additional robustness, particularly in multi-task scenarios such as QA-SA-Summ.

Interestingly, while SSR performs better than EVCL-SSR on prior tasks, the inclusion of earlier training data during SSR reduces the effectiveness of training on new tasks. This can result in marginally lower performance for current tasks. However, EVCL, through its Bayesian inference framework, better balances task-specific plasticity and stability, encapsulating new training data without heavily compromising prior knowledge.

The results underscore the scalability and adaptability of EVCL-SSR, particularly with larger models like the 8B configuration, where catastrophic forgetting is more prevalent. In smaller models like the 3B, the differences in performance between methods are less pronounced, indicating the robustness of both approaches in resource-constrained settings.

5.2 Comparison of SSR to LoRA

SSR's performance was found to be comparable to that of LoRA across configurations, with some variability in task-specific outcomes. For instance, in the 8B model QA configuration, both SSR and LoRA achieved identical scores of 15.06, demonstrating parity in performance for simpler tasks. However, for more complex tasks such as QA-SA-Summ, SSR (average ROUGE-L: 26.34) slightly outperformed LoRA (average ROUGE-L: 18.36), reflecting a notable improvement of 7.98 points.

In the 3B model, SSR and LoRA showed mixed trends. For QA-SA-Summ, SSR scored 18.95, slightly higher than LoRA's score of 15.42, indicating better performance of SSR in retaining task knowledge during continual learning. These results reinforce SSR's viability as an alternative to LoRA, particularly in continual learning contexts where its internal data generation reduces reliance on external datasets.

5.3 EVCL Performance

EVCL consistently outperformed both SSR and LoRA 2, with significant improvements for certain configurations. For example, in the 8B model QA-SA-Summ configuration, EVCL achieved an average ROUGE-L score of 28.37, surpassing SSR's 26.34 and LoRA's 18.36 by 2.03 and 10.01 points, respectively. Similarly, in the 3B model QA-SA-Summ configuration, EVCL scored 18.59, closely matching SSR's 18.95 but outperforming LoRA's 15.42 by 3.17 points.

These improvements demonstrate EVCL's ability to effectively balance learning stability and plasticity. The marked edge provided by EVCL, particularly in multi-task scenarios, highlights its potential in mitigating catastrophic forgetting (CF) while maintaining adaptability across tasks.

5.4 Model Size and Catastrophic Forgetting

A significant observation from our experiments was the disparate levels of catastrophic forgetting exhibited by the two models of varying sizes. The larger LLaMA 3 (8B) model, despite its higher parameter count and capacity, demonstrated greater difficulty in retaining knowledge across tasks, as evidenced by lower backward transfer (BWT) scores compared to the smaller LLaMA 3.2 (3B) model. For instance, in the QA-SA-Summ configuration, the 8B model's BWT for SSR was only 1.14, markedly lower than the 3.2B model's 1.67. This suggests that larger models may be more prone to overwriting previously learned information.

Interestingly, the larger 8B model was more effective at generating high-quality synthetic data for SSR, an advantage likely stemming from its greater learning capacity and depth. This capability, however, did not translate into better task retention. In contrast, the smaller 3B model, while not as proficient in generating synthetic data, showed better overall task retention,

Table 1: Llama 3 (8B) Results (Values Scaled by 100) - Average ROUGE-L and BWT scores across three fine-tuning methods: 1) LoRA-based PEFT 2) SSR-EVCL (Our Method) 3)SSR Only. The results demonstrate improved accuracy with our method, reflected in higher average ROUGE-L scores.

Method	**QA**	**SA**	**Summ**	**Avg Rouge L**	**BWT**
LoRA-based PEFT QA	15.06			15.06	
LoRA-based PEFT QA-SA	11.234	34.01		22.622	7.562
LoRA-based PEFT QA-SA-Summ	10.54	29.11	15.43	18.36	4.262
SSR-EVCL-Our Method QA	13.66			13.66	
SSR-EVCL-Our Method QA-SA	13.6	59.08		**36.34**	22.68
SSR-EVCL-Our Method QA-SA-Summ	12.82	55.64	16.65	**28.37**	3.97
SSR Only QA	15.06			15.06	
SSR Only QA-SA	12.11	38.29		25.2	10.14
SSR Only QA-SA-Summ	11.78	51.24	16.01	26.34	1.14

Table 2: LLaMA 3.2 (3B) Results (Values Scaled by 100) - Average ROUGE-L and BWT scores across three fine-tuning methods: 1) LoRA-based PEFT, 2) SSR-EVCL (Our Method), and 3) SSR Only. The results indicate that while the effects of catastrophic forgetting are less clear for the 3B model, our method achieves comparable performance.

Method	**QA**	**SA**	**Summ**	**Avg Rouge L**	**BWT**
LoRA-based PEFT QA	6.52			6.52	
LoRA-based PEFT QA-SA	6.51	19.74		13.12	6.6
LoRA-based PEFT QA-SA-Summ	6.36	19.79	20.13	15.42	2.3
SSR-EVCL-Our Method QA	8.34			6.34	
SSR-EVCL-Our Method QA-SA	8.68	23.31		**15.99**	9.65
SSR-EVCL-Our Method QA-SA-Summ	7.92	25.48	22.38	18.59	1.78
SSR Only QA	6.52			6.52	
SSR Only QA-SA	6.813	21.54		14.23	7.50
SSR Only QA-SA-Summ	6.46	22.47	21.78	**18.95**	1.67

likely due to more efficient parameter utilization and less susceptibility to catastrophic forgetting.

These findings highlight the importance of robust continual learning strategies, such as EWC and our proposed SSR-EVCL framework, particularly for managing the complexities of large-scale models like the 8B configuration. The SSR-EVCL framework specifically addresses these issues by integrating efficient fine-tuning and synthetic data refinement processes that help improve both models' ability to maintain knowledge across multiple tasks.

5.5 Novel Application of EVCL in NLP

One of the key contributions of this work is the application of EVCL to NLP tasks, a domain where it has not been extensively explored. EVCL demonstrated clear advantages, particularly in mitigating CF for larger models. For instance, in the 8B model QA-SA-Summ configuration, EVCL achieved an average ROUGE-L score of 28.37, significantly outperforming both SSR (26.34) and LoRA (18.36). For the 3B model, EVCL achieved 18.59, closely trailing SSR (18.95) but still exceeding LoRA (15.42).

These results suggest that while EVCL's parameter uncertainty modeling provides benefits, its practical utility in NLP tasks can be further optimized. When combined with iterative synthetic rehearsal mechanisms, EVCL offers a promising pathway for enhancing task retention and mitigating CF in real-world continual learning applications.

6 Discussion

The results reveal that EVCL provides improvements over SSR and Lora in mitigating catastrophic forgetting, particularly in multi-task scenarios. However, these gains are incremental, and SSR remains a competitive alternative, especially given its simplicity and data efficiency. Larger models like the 8B configuration exhibit more catastrophic forgetting than smaller models like the 3B, emphasizing the need for robust continual learning methods in large-scale applications. Finally, the application of EVCL to NLP tasks, while novel, yielded mixed results, demonstrating its potential but also indicating the need for further exploration and refinement.

7 Author Contributions

The project comprised two core components: **Self-Synthesized Rehearsal (SSR)** and **Elastic Variational Continual Learning (EVCL)**, with equal contributions from all authors toward the research. Each component was further divided into subtasks, which were assigned as follows:

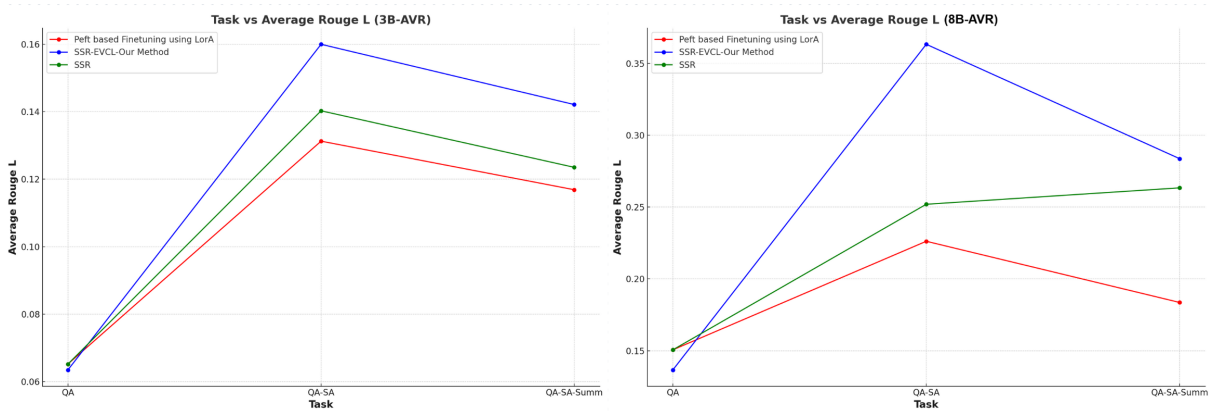

Figure 2: Tasks vs. ROUGE-L Scores Plot (Left: LLaMA 3.2 (3B), Right: LLaMA 3 (8B)). This figure highlights the superior ROUGE-L scores achieved by our model for both configurations, demonstrating improved output accuracy and reduced catastrophic forgetting through continual learning, outperforming standard fine-tuning methods.

7.1 Self-Synthesized Rehearsal (SSR)

7.1.1 Synthetic Data Generation

Vibhor Tyagi and Shreya Shetye collaboratively implemented the data generation pipeline using LLaMA 3 8B and 3.2 3B models. Vibhor focused on designing in-context learning prompts to guide the creation of synthetic context-question pairs, while Shreya optimized the generation process, including model configuration for 8-bit precision and hyperparameter tuning.

7.1.2 Data Refinement

Vibhor and Shreya jointly developed the clustering process using K-Means for grouping synthetic data and implemented representative subset sampling to ensure diversity and reduce redundancy. Shreya handled the iterative feedback loop with the Prometheus evaluation model to refine synthetic instances, while Vibhor ensured the refinement process aligned with task-specific quality metrics.

7.1.3 SSR Evaluation

Shreya and Vibhor shared responsibility for evaluating the quality of refined rehearsal data. Shreya focused on measuring its task relevance and coherence, while Vibhor analyzed its impact on task retention and mitigation of catastrophic forgetting.

7.2 Elastic Variational Continual Learning (EVCL)

7.2.1 EVCL Implementation

Pranav Sharma and Yash Samir Kakde jointly implemented the EVCL framework. Pranav focused on integrating LoRA fine-tuning and Fisher Information Matrix (FIM) computation for parameter importance, while Yash designed the variational posterior initialization and Bayesian parameter updates.

7.2.2 Training Pipeline

Pranav and Yash collaboratively developed the training pipeline. Pranav configured hyperparameters such as learning rate, epochs, and batch size for both LLaMA 3 8B and 3.2 3B models, while Yash integrated the EWC regularization term into the variational loss function and ensured compatibility with LoRA's parameter updates.

7.2.3 EVCL Evaluation

Pranav and Yash shared responsibility for evaluating EVCL performance. Pranav analyzed task retention and mitigation of catastrophic forgetting, while Yash focused on the balance between stability and plasticity and its generalization capabilities across sequential tasks.

7.3 Collaborative Efforts

All team members contributed equally to the integration of SSR and EVCL, ensuring a seamless interaction between the synthetic data rehearsal mechanism and the continual learning framework. They worked together on result interpretation, preparing figures and tables, and drafting the manuscript, reflecting equal contributions to the final outcome.

References

[Bib2024] 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models, December. [Online; accessed 9. Dec. 2024].

[Bingham et al.2019] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. 2019. Pyro: Deep universal probabilistic programming. *Journal of machine learning research*, 20(28):1–6.

[Dubey et al.2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[Hu et al.2021] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[Huang et al.2024] Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *arXiv preprint arXiv:2403.01244*.

[Kirkpatrick et al.2017] 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.

[Li et al.2024] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*.

[Lin2004] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[Luo et al.2023] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

[Nguyen et al.2017] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. 2017. Variational continual learning. *arXiv preprint arXiv:1710.10628*.

[Rostami et al.2019] Mohammad Rostami, Soheil Kolouri, and Praveen K Pilly. 2019. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*.

[Shlens2014] Jonathon Shlens. 2014. Notes on kullback-leibler divergence and likelihood. *arXiv preprint arXiv:1404.2000*.

[Wang et al.2022] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.