Improving Continual Pre-training Through Seamless Data Packing

Ruicheng Yin[*], **Xuan Gao**[*], **Changze Lv, Xiaohua Wang,**
Xiaoqing Zheng[†]**, Xuanjing Huang**
School of Computer Science, Fudan University, Shanghai, China
{rcyin23,gaox23}@m.fudan.edu.cn
{zhengxq,xjhuang}@fudan.edu.cn

Abstract

Continual pre-training has demonstrated significant potential in enhancing model performance, particularly in domain-specific scenarios. The most common approach for packing data before continual pre-training involves concatenating input texts and splitting them into fixed-length sequences. While straightforward and efficient, this method often leads to excessive truncation and context discontinuity, which can hinder model performance. To address these issues, we explore the potential of data engineering to enhance continual pre-training, particularly its impact on model performance and efficiency. We propose **Seamless Packing** (SP), a novel data packing strategy aimed at preserving contextual information and enhancing model performance. Our approach employs a sliding window technique in the first stage that synchronizes overlapping tokens across consecutive sequences, ensuring better continuity and contextual coherence. In the second stage, we adopt a First-Fit-Decreasing algorithm to pack shorter texts into bins slightly larger than the target sequence length, thereby minimizing padding and truncation. Empirical evaluations across various model architectures and corpus domains demonstrate the effectiveness of our method, outperforming baselines in 99% of all settings. Code is available at https://github.com/Infernus-WIND/Seamless-Packing.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in various natural language understanding and generation tasks (Brown et al., 2020; Touvron et al., 2023; Yang et al., 2024a). These models are usually pre-trained on extensive corpora spanning diverse domains (Radford and Narasimhan, 2018), enabling them to perform a wide range of applications. Recent studies have shown that continual pre-training is an effective strategy for adapting LLMs to specific domains or tasks (Chen et al., 2024; Jang et al., 2022; Gupta et al., 2023), bridging the gap between general-purpose pre-training and task-specific fine-tuning (Biesialska et al., 2020; Mehta et al., 2023).

Continual pre-training shares many similarities with standard pre-training, particularly in data packing methods, which involve splitting training data into fixed length sequences to facilitate parallelized training. However, while significant attention has been given to domain-specific data selection, sampling strategies, and mitigating catastrophic forgetting in continual pre-training (Lesort et al., 2021; Winata et al., 2023; Parmar et al., 2024), relatively little research has focused on the effective organization of training data within batches.

Currently, the most widely adopted method for data packing is straightforward: concatenating all texts into a single stream and segmenting it into fixed-length sequences based on the model's input size (Gururangan et al., 2020). While efficient and widely used, this approach has a notable drawback—arbitrary truncation of sequences within one piece of text often disrupts contextual continuity, resulting in incomplete sequences. Such disruptions can degrade the model's ability to understand and generate coherent text, potentially leading to performance issues such as hallucination in downstream tasks (Ding et al., 2024; Achiam et al., 2023). Some existing data packing methods instead rely on padding to ensure uniform sequence lengths. However, padding introduces its own inefficiencies—it occupies valuable input space without any meaningful information, reducing the proportion of real data processed per step. This not only limits training efficiency but may also weaken the model's ability to leverage longer-range dependencies when a substantial portion of input sequences consists of meaningless tokens.

---

[*] These authors contributed equally.
[†] Corresponding author.

To address these challenges, we frame the data packing process as an optimization problem aimed at maximizing the contextual continuity within sequences while minimizing truncation and padding, subject to constraints on the pre-defined sequence length and maximum overlapping contexts.

We propose **Seamless Packing** (SP) as a solution. To optimize sequence utilization, our method follows a two-stage process with a sequential order. The first stage prioritizes long texts that meet specific criteria, dynamically generating overlapping sequences while maintaining a predefined maximum repetition ratio to avoid excessive redundancy. In conventional data packing, the final portion of a long text often fails to fill a complete sequence, requiring concatenation with subsequent texts. This results in fragmented chunks with insufficient context and additional truncation of the following text. By contrast, the first stage of our method maximizes contextual continuity and minimizes truncation by avoiding such fragmentation, enabling a seamless allocation of sequences, where a long text can fully occupy multiple consecutive sequences without requiring fragments from other texts.

In the second stage, those shorter texts left behind by first stage are packed using an approximation algorithm, First-Fit-Decreasing (FFD). The packing bins are assigned a capacity slightly larger than the sequence length to reduce padding, while any tokens exceeding the sequence length are discarded to maintain efficiency. This stage ensures a seamless arrangement of text by minimizing both padding and truncation. These two stages contribute to seamless text packing from different angles, preserving contextual continuity. By integrating them, our method effectively addresses the optimization problem, resulting in an effective and systematic solution for data packing in continual pre-training. Furthermore, our method complements and can be integrated with existing approaches that focus on data sampling or mitigating catastrophic forgetting, as it targets a distinct yet important aspect of continual pre-training—optimizing data packing.

To evaluate the effectiveness of our method, we conduct comprehensive experiments on a variety of models and domains. The results demonstrate that our method consistently outperforms conventional data packing methods, achieving superior performance and generalization across diverse settings. The main contributions are summarized as follows:

- We propose **Seamless Packing**, a simple yet effective data packing method tailored for continual pre-training. By optimizing both segment placement and overlap strategy, our approach preserves *contextual continuity* while reducing truncation and padding.
- We present a theoretical analysis of Seamless Packing, characterizing the influence range of overlapping contexts under the control of two key hyperparameters.
- We conduct extensive experiments across diverse domains and tasks to validate the effectiveness and generalizability of our method. Results show consistent improvements across domain-specific and general-domain settings.
- Through ablation studies and controlled comparisons, we validate the contribution of each component of our method. We also explore the trade-off between token dropping and padding, and demonstrate how Seamless Packing mitigates truncation-induced hallucination by preserving critical contextual information through a targeted case study.

2 Related Work

Continual Pre-training Continual pre-training has been shown to effectively enhance language models by adapting them to specific domains or improving their general capabilities (Gururangan et al., 2020; Yıldız et al., 2024). This approach involves a second phase of pre-training on unlabeled data, allowing models to integrate new knowledge without requiring extensive labeled datasets.

Data Packing Several studies have explored data packing strategies to optimize language model training. Krell et al. (2021) proposed a packing algorithm designed to maximize sequence length utilization, thereby reducing the need for padding tokens. Ding et al. (2024) conducted both theoretical and empirical analyses on the impact of truncation, emphasizing that preserving data integrity is crucial for improving language modeling performance. To address this, they introduced a Best-Fit-Decreasing packing algorithm based on segment trees, which optimally reduces truncation. Our proposed method integrates the strengths of these approaches while introducing novel innovations, achieving a balanced optimization of the data packing process that preserves contextual continuity and minimizes truncation and padding, leading to a seamless packing algorithm. More information is provided in Appendix A.
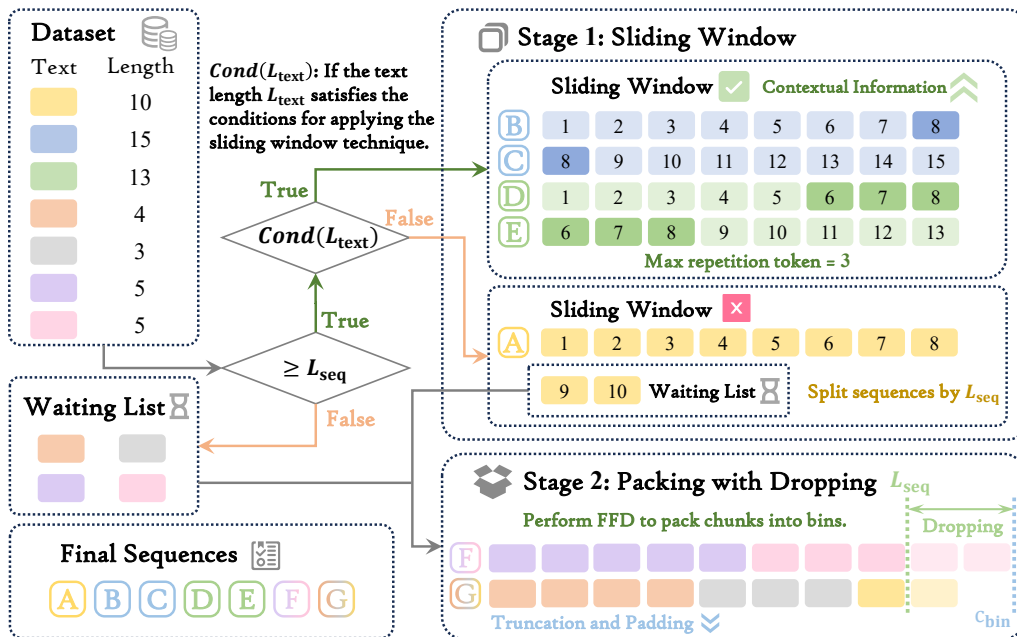
Figure 1: An illustration of the proposed Seamless Packing method. **Top-right**: Stage 1, each text is assessed to determine whether the sliding window technique can be applied. **Top-right (upper)**: Sliding window applicable—darker-colored tokens indicate repeated tokens. **Top-right (lower)**: Sliding window not applicable—chunks that cannot fill an entire sequence are added to the waiting list. **Bottom-right**: Stage 2, short chunks are packed using the First-Fit-Decreasing strategy. Lighter-colored tokens represent discarded tokens. Note: L_{seq} - sequence length, c_{bin} - bin capacity. Both L_{seq} and c_{bin} are hyperparameters, which are set to $L_{seq} = 8$ and $c_{bin} = 10$ in this figure.

3 Method

In this section, we introduce Seamless Packing in detail. As illustrated in Figure 1, our method consists of two stages: Sliding Window and Packing with Dropping. The design of the two stages is motivated by the need to address two crucial challenges in data packing for continual pre-training: preserving contextual continuity and minimizing truncation and padding, the necessity of which has been highlighted in previous section. In the first stage, long texts that meet certain criteria are processed using a dynamic sliding window technique, effectively leveraging contextual continuity and minimizing truncation. The second stage processed the remaining short texts, reducing both truncation and padding. Both stages are carefully designed to ensure the algorithm is seamless and improve the overall performance of LLMs in continual pre-training scenarios.

3.1 The Importance of Contextual Continuity

Before introducing our packing strategy, we first examine how naive truncation—commonly used in continual pre-training pipelines—can disrupt contextual continuity. While efficient, this preprocessing approach often fragments semantically connected content, undermining the model's ability to reason over longer contexts.

First, consider a sentence like *"Event A is [DE-SCRIPTION], which will be held on [DATE], in [PLACE]."* If the input is segmented just before "which", the event description remains in one chunk, while time and location details appear in another. This disconnection can hinder the model's ability to associate the components of a single event.

Second, in summarization tasks, truncating key explanatory context may result in unfaithful outputs. For example, if *"The Supreme Court ruled in favor of the plaintiff after considering key arguments related to privacy law"* is split before "after considering...," the generated summary may omit or hallucinate the basis for the ruling.

Third, in code understanding, a variable declaration split from its usage—e.g., when the definition appears in one segment and its reference in another—can lead to incomplete or incorrect completions, as dependencies are no longer visible (Ding et al., 2024).

These examples underscore the importance of preserving contextual continuity in data preprocessing. To address this, our Seamless Packing

method is designed to minimize context fragmentation through overlapping windows and adaptive sequence packing. In Section 4.7, we further support this analysis with a case study that empirically demonstrates the detrimental effects of truncation.

3.2 First Stage: Sliding Window

The first stage of our method is inspired by the sliding window technique (Boyer and Moore, 1977; Karp and Rabin, 1987; Koç, 1995), which is widely employed in algorithm design across various fields. Traditional sliding window uses a fixed stride to determine how much the window advances at each step. However, a fixed stride is suboptimal in Seamless Packing. *Fixed* strides cannot dynamically adapt to varying texts lengths, leading to either incomplete coverage or excessive overlap, which disrupts seamless alignment of sequences. Moreover, an overly small stride results in a high overlap ratio between consecutive sequences, reducing training efficiency and potentially degrading model performance, as observed in prior studies (Allamanis, 2019; Lee et al., 2022). Conversely, larger stride values complicate the intuitive understanding of the sliding window's purpose in our method, which is to facilitate context sharing across sequences.

To overcome these limitations, we introduce a more intuitive parameter, maximum repetition ratio, denoted as $r_{\max}$, which offers a more precise control over sequence overlap, ensuring optimal training dynamics and model efficacy. Given a tokenized text of length L_{original} that can be divided into n full sequences of length L_{seq}, the maximum number of overlapping tokens allowed is:

$$L_{\text{max\_overlap}} = \lceil n \times r_{\max} \times L_{\text{seq}} \rceil. \tag{1}$$

The sliding window technique is applicable when:

$$L_{\text{original}} + L_{\text{max\_overlap}} \geq (n+1) \times L_{\text{seq}}, \tag{2}$$

which ensures that the text can be expanded to fill $n + 1$ sequences. In such cases, the overlap is dynamically adjusted to achieve full sequence utilization:

$$L_{\text{final\_overlap}} = \lceil \frac{(n+1) \times L_{\text{seq}} - L_{\text{original}}}{n} \rceil. \tag{3}$$

This strategy ensures an efficient balance between overlap and utilization of available sequence space.

For texts that do not satisfy the condition described in expression (2), we simply divide them into chunks of length L_{seq}. Chunks that entirely occupy a sequence are retained, while incomplete

---

Algorithm 1: First-Fit-Decreasing

Input: Texts $T = \{t_i\}_{i=1}^N$
Output: Bins $\{b_j\}$ containing all items
Define $l(t)$: length of text t;
Define $rc(b)$: remaining capacity of bin b;
Sort T in descending order of length, resulting in T_s;
Initialize an empty set of bins: $\{b_i\}_{i=1}^N$;
foreach $t_i \in T_s$ **do**
    Find the first bin b_j such that
    $rc(b_j) \geq l(t_i)$;
    Add t_i to b_j;

---

chunks are deferred to the second stage for further processing.

The advantages of our sliding window technique are twofold. First, it allows some sequences to incorporate additional context compared to the traditional method. Second, it reduces cases where chunks from different documents are concatenated within a single sequence. While such cases are handled by the model through adding special token or attention masking (Vaswani et al., 2017; Devlin et al., 2019), the contextual information of chunks in such situation is often limited, offering minimal benefit to training. By mitigating these issues, our method enhances context preservation and ultimately improves model performance.

3.3 Second Stage: Packing with Dropping

The first stage has processed all chunks that can occupy an entire sequence, leaving short chunks behind. Thus, the main purpose of second stage is to pack these short chunks together efficiently to maximize sequence utilization.

At its core, this stage is formulated as a variant of the bin packing problem (Martello and Toth, 1990), which seeks to minimize the number of fixed-capacity bins required to accommodate items of varying sizes. As this problem is NP-hard, we employ an approximation algorithm, specifically the First-Fit-Decreasing (FFD) (Johnson, 1973) heuristic. While Best-Fit-Decreasing (BFD) (Eilon and Christofides, 1971) is another widely used heuristic, we adopt FFD because its early stopping mechanism significantly reduces computation time while achieving competitive performance, as we will show in Section 4.5.

While the FFD algorithm we employ follows a standard implementation, we introduce a flexi-

ble modification to better adapt to different dataset characteristics. Specifically, we allow the bin capacity to be slightly larger than L_{seq}, controlled by a parameter called extra bin capacity, denoted as c_{extra}, and tokens that exceed L_{seq} after packing are discarded. This adjustment provides an alternative strategy rather than strictly relying on padding, making the approach adaptable to different scenarios. While padding could be used as an alternative under certain conditions, we firmly adopt the dropping strategy in this study and demonstrate its effectiveness across all experimental settings. Our results show that dropping consistently achieves competitive performance, reinforcing its suitability as the preferred choice.

The rationale behind this design is twofold. First, dropping eliminates the inefficiencies associated with excessive padding, ensuring that training is not diluted by meaningless tokens and maintains a seamless algorithm. Second, the number of discarded tokens can be precisely controlled by c_{extra}, and when properly tuned, the fraction of dropped tokens remains negligible, having minimal impact on overall performance. The complete FFD algorithm we used is outlined in Algorithm 1.

After applying the above procedure, a small number of bins may still contain sequences whose total length shorter than L_{seq}. At this point, the remaining data volume is minimal. To ensure that Seamless Packing remains a padding-free method, we concatenate all remaining chunks into a single sequence, which is then divided into chunks of length L_{seq} and added to the final output.

The primary objective of the Packing with Dropping approach is to process shorter text chunks while avoiding unnecessary truncation or padding, both of which have been previously identified as limitations. By minimizing these inefficiencies, this stage further contributes to improving the overall performance of the model.

3.4 Theoretical Analysis

We now conduct a theoretical analysis of the proportion of texts that satisfy the condition in expression (2) and the amount of shorter chunks processed in stage two, as influenced by r_{max}. This analysis allows us to understand the proportion of data affected by each stage of our method, providing a more comprehensive view of its impact.

Before delving into the analysis, we first examine the dataset's length distribution. We present the token count distribution of datasets we used in

Appendix C. A key observation is that the number of texts within each length interval decreases as the sequence length increases. While the actual distribution is unlikely to follow a perfectly linear trend, the samples within each interval can be reasonably approximated as uniformly distributed for analytical purposes.

By combining expressions (1) and (2), we derive the condition for applying the sliding window:

$$L_{\text{original}} \geq (n + 1 - nr) \times L_{\text{seq}}. \tag{4}$$

Given the definition of n, we also have:

$$L_{\text{original}} \geq n \times L_{\text{seq}}. \tag{5}$$

Since expression (5) always holds, if $n + 1 - nr \leq n$ is satisfied, expression (4) follows directly. This implies that when $n \geq \lceil \frac{1}{r} \rceil$, all relevant texts can be processed using the sliding window.

For intervals where $n < \lceil \frac{1}{r} \rceil$, the proportion of text samples within each interval that can be processed using the sliding window is nr. Therefore, the total number of texts N_{sw} that can be processed using the sliding window is given by:

$$N_{sw} = \sum_{k=1}^{\lceil \frac{1}{r} \rceil} (krT_k) + \sum_{k=\lceil \frac{1}{r} \rceil}^{M} T_k, \tag{6}$$

which evidently increases as r increases and T_k denotes the number of texts whose tokenized length falls within the interval $(kL_{\text{seq}}, (k+1)L_{\text{seq}}]$, where $k \in \{1, 2, \ldots, M\}$ and M represents the largest interval considered in the dataset.

Next, we consider the texts that cannot be processed using the sliding window, which are divided into shorter chunks. The total number of tokens for these short texts is given by:

$$N_{\text{token\_short}} = \sum_{k=1}^{\lceil \frac{1}{r} \rceil} \left[(1 - kr)T_k \times \frac{(1 - kr)L_{\text{seq}}}{2} \right]. \tag{7}$$

In this expression, the first part, $(1 - kr)T_k$, represents the number of short texts within each interval that cannot be processed using the sliding window. The second part, $\frac{(1-kr)L_{\text{seq}}}{2}$, is the average length of the short texts in each interval, assuming a uniform distribution of lengths within the interval. Thus, the formula computes the total number of tokens in all these short texts. We further illustrate our theoretical analysis in appendix B.

Tasks	GPT2 - large (812M)				Llama3.2 - 1B				Qwen2.5 - 1.5B			
	OM	CT	BFD	SP	OM	CT	BFD	SP	OM	CT	BFD	SP
BBC News	97.18	96.80	97.22	**97.38**	97.22	97.44	97.54	**97.64**	97.68	97.74	97.80	**98.06**
AG News	89.29	88.66	88.32	**89.65**	89.76	89.58	89.97	**90.00**	88.53	89.39	89.71	**90.37**
20 Newsgroup	65.76	66.35	65.60	**67.13**	65.73	64.74	65.71	**66.30**	67.11	67.78	67.24	**68.02**
Fin Topic	81.39	81.71	81.79	**81.99**	82.84	82.69	**84.62**	84.58	81.99	83.13	82.69	**84.18**
Fin Sentiment	85.33	86.79	87.08	**87.29**	87.25	88.18	88.18	**88.68**	86.92	88.51	88.68	**88.85**
Fin Phrasebank	82.78	82.10	82.94	**83.04**	82.06	83.96	83.70	**84.44**	84.08	84.02	84.12	**84.52**
PubMed Class	84.62	84.69	85.17	**85.88**	83.09	84.18	83.67	**84.79**	85.10	86.32	86.22	**86.87**
ChemProt	80.96	81.48	81.51	**82.35**	77.71	80.03	**80.81**	**80.81**	81.39	81.89	81.83	**82.64**

Table 1: Performance on downstream tasks after full parameter fine-tuning, comparing original pre-trained model and three different data packing methods used in continual pre-training. Note: OM - Original Model, CT - Concatenation and Truncation, BFD - Best Fit Decreasing, SP - Seamless Packing, Fin - Financial, class - classification

4 Experiments

We conduct extensive experiments to assess the effectiveness of Seamless Packing across diverse domains and models. Unless otherwise specified, models are independently pre-trained on domain-specific corpora. Additionally, we design ablation studies to validate the contributions of both stages in our method. Through comprehensive evaluations, we analyze the impact of hyperparameters, validate the theoretical analysis and further investigate key aspects of our method. The results demonstrate that our method consistently outperforms traditional approaches across all tasks, achieving a fourfold improvement over baseline methods (0.96% vs. 0.24%). All fine-tuning results are averaged over five runs (seeds 42–46).

4.1 Datasets and Baselines

The dataset used for continual pre-training are BBC news (Li et al., 2024), financial news article (Jeet.J, 2018) and academic articles from PubMed (Cohan et al., 2018). Our experiments are conducted on GPT-2 (Radford et al., 2019), LLaMA-3 (Dubey et al., 2024), Qwen2.5 (Yang et al., 2024b) and Gemma-2 (Team et al., 2024). The universal baselines consist of three configurations: the original model, the model continual pre-trained through traditional concatenation and truncation, and the model continual pre-trained using BFD algorithm. More details are presented in Appendix D.

4.2 Main Results

Continual pre-training perplexity Perplexity is a widely used evaluation metric in natural language processing to measure the quality of a language model's predictions (Jelinek et al., 1977;

Model	Domain	CT	BFD	SP
GPT2 812M	News	10.79	10.28	**10.13**
	Finance	7.19	6.74	**6.73**
	Med	11.49	10.16	**10.12**
Llama3.2 1B	News	11.48	10.52	**10.07**
	Finance	6.27	5.71	**5.52**
	Med	7.84	7.41	**7.39**
Qwen2.5 1.5B	News	9.51	9.20	**9.19**
	Finance	4.76	4.56	**4.50**
	Med	6.71	6.47	**6.43**

Table 2: Perplexity on the validation dataset during continual pre-training, comparing three different data packing methods on three models and three domains respectively.

Miaschi et al., 2021). In the context of continual pre-training, perplexity serves as a key indicator of the effectiveness of the training approach. As shown in Table 2, our proposed method achieves a significantly lower perplexity on the validation set compared to the baseline approach, demonstrating its superior ability to capture linguistic structure and generalize to unseen data. This improvement underscores the effectiveness of Seamless Packing in leveraging overlapping contexts to enhance the model's understanding and predictive accuracy.

Full parameter fine-tuning We evaluate the performance of our method on eight diverse downstream tasks spanning the three selected domains. The models are fine-tuned on downstream tasks after continual pre-training using different methods. The results are presented in Table 1 and statistical significance tests are summarized in Appendix E.

The results show that while performance gains vary across models and tasks, our method consistently achieves the best results in the majority of

Model	Metric/Task	CT	BFD	SP
Qwen2.5 3B	Perplexity	10.27	9.40	**9.24**
	BBC News	97.54	**97.56**	97.50
	AG News	88.13	87.21	**88.66**
	20 Newsgroup	58.65	58.95	**59.38**
Gemma2 2B	Perplexity	11.11	**10.97**	11.02
	PubMed Class	85.03	84.86	**85.10**
	ChemProt	65.98	63.95	**66.47**

Table 3: Perplexity on the validation dataset during continual pre-training and performance on downstream tasks after LoRA tuning, comparing three different data packing methods.

cases. This highlights not only the strong efficacy of our approach but also its robustness across different domains and task types with diverse linguistic complexity and data distributions.

Additionally, our method exhibits stable gains in scenarios where baseline methods struggle. For every model, BFD baseline method under-performs traditional method of concatenation and truncation in 2 to 4 tasks, whereas our method maintains robust performance across all tasks without such failures. This suggests that our improvements are both significant and broadly applicable across different task characteristics.

Overall, the empirical results support our hypothesis that our method optimizes contextual understanding during continual pre-training, leading to better performance. The consistent improvements across domains further confirm its scalability.

LoRA tuning LoRA (Low-Rank Adaptation, Hu et al., 2022) is a widely adopted parameter-efficient fine-tuning (PEFT) method that updates only a small subset of parameters, making it particularly effective for adapting large pre-trained models to specific tasks with minimal computational overhead. We include this set of experiments to demonstrate that our method not only excels under full-parameter fine-tuning but also achieves strong performance with PEFT approaches such as LoRA. As shown in Table 3, our method consistently achieves competitive results under LoRA tuning, further highlighting its adaptability to mainstream fine-tuning paradigms.

4.3 Generalization Analysis

To assess the robustness and adaptability of Seamless Packing beyond specific data configurations, we conduct experiments under two distinct settings: mixed-domain and general-domain.

Model	Task	CT	BFD	SP
Llama3.2 1B	20 Newsgroup	68.72	68.55	**69.80**
	Fin Phrasebank	84.10	84.04	**84.58**
	ChemProt	81.22	82.14	**82.58**

Table 4: Evaluation results of mixed-domain continual pre-training.

Model	Task	CT	BFD	SP
GPT2 812M	20 Newsgroup	69.15	69.42	**69.69**
	ChemProt	81.62	80.64	**82.50**
	MNLI	68.95	67.93	**70.47**
	QNLI	80.77	80.40	**81.80**
	RTE	69.53	69.31	**71.33**

Table 5: Evaluation results of general-domain continual pre-training.

Mixed-Domain We pre-train a model on a mixed-domain corpus that combines three specialized domains: News, Finance, and Medical. Evaluation is conducted on three downstream tasks, each corresponding to one of the original domains. This setup enables us to test the model's ability to retain and integrate heterogeneous domain-specific knowledge. Table 4 reports the performance comparison among different packing strategies.

General-Domain To examine whether SP generalizes beyond domain-specific settings, we adopt RedPajama (Weber et al., 2024), an open-source dataset designed to approximate the original LLaMA training corpus. To avoid overlap with the original LLaMA training data, we select GPT-2-Large as the base model and conduct continual pre-training on a sampled subset of RedPajama. The model is evaluated on two domain-specific downstream tasks (20 Newsgroup, ChemProt) to assess cross-domain transferability, and on three general-purpose tasks from the GLUE benchmark (Wang et al. (2018), MNLI, QNLI, RTE) to evaluate natural language understanding capabilities. Results are shown in Table 5.

Discussion Across both experiments, SP consistently achieves superior performance. Whether operating in mixed-domain settings or pre-training from general-domain corpora, our method outperforms both naive concatenation and standard bin-packing strategies. These findings confirm that SP generalizes effectively across both domain-specific and general-domain continual pre-training, supporting its applicability in a wide range of real-world scenarios.

Model	Task	OM	CT	BFD	SP
Llama3.2 1B	XNLI (French)	63.87	67.72	68.64	**69.04**

Table 6: Evaluation results of the French XNLI task.

Metric/Task	Model	FFD	BFD	BFD-m	SP
Perplexity	Qwen2.5 0.5B	**11.24**	**11.24**	**11.24**	**11.24**
	Llama3.2 1B	12.08	12.39	**11.95**	11.99
	Qwen2.5 1.5B	9.45	**9.39**	9.40	9.48
PubMed Class	Qwen2.5 0.5B	85.17	85.10	85.23	**86.12**
	Llama3.2 1B	85.00	84.89	85.37	**85.47**
	Qwen2.5 1.5B	85.78	85.85	86.11	**86.56**
ChemProt	Qwen2.5 0.5B	78.78	78.64	79.07	**80.26**
	Llama3.2 1B	80.38	80.38	80.44	**81.45**
	Qwen2.5 1.5B	81.83	82.00	82.09	**82.53**

Table 7: Perplexity on the validation dataset during continual pre-training and performance on downstream tasks after fine-tuning, comparing four different data packing methods. The underlined text represents the second-best performance.

4.4 Cross-lingual Transfer

To further evaluate the cross-lingual applicability of Seamless Packing, we conduct experiments on a French dataset. Specifically, we continually pre-train Llama-3.2-1B on a sampled French subset of the C4 corpus (Raffel et al., 2020) and evaluate the model on the French NLI task from the FLUE benchmark (Le et al. (2019), French Language Understanding Evaluation). The XNLI dataset used here provides a direct measure of a model's ability to perform natural language inference in a non-English language. Table 6 presents the result and our method again yields the highest performance.

These results demonstrate that Seamless Packing is effective even in cross-lingual scenarios, highlighting its potential to support multilingual continual pre-training and downstream task adaptation.

4.5 Ablation Study

We conduct a series of ablation studies to evaluate the contributions of the two key stages in Seamless Packing and to compare BFD and FFD. We select three different models and continually pre-train them on a resampled PubMed article dataset using various data packing methods. We verify model's performance on two downstream tasks with full parameter fine-tuning. Additionally, we introduce two alternative methods: FFD and BFD-m. FFD employs the FFD algorithm to pack all texts into bins with a capacity of L_{seq}, whereas BFD-m extends BFD by allowing a larger bin capacity, ex-

ceeding L_{seq}. BFD-m serves as an intermediate variant to isolate effects of two stages in Seamless Packing: comparing BFD with BFD-m reveals contribution of the second stage, while comparing BFD-m with SP highlights impact of the first stage.

As shown in Table 7, although Seamless Packing does not always achieve the lowest perplexity, it consistently delivers the best downstream task performance in most cases, outperforming BFD-m. This result highlights the effectiveness of the sliding window technique in our approach. Furthermore, BFD-m outperforms standard BFD in all scenarios, suggesting that slightly increasing bin capacity enhances overall model performance.

Interestingly, FFD performs comparably to BFD on average, indicating that BFD may be unnecessary in scenarios where it does not provide a clear performance advantage but incurs higher computational costs. A detailed time consumption analysis is provided in Section 4.8.

We also conducted experiments to evaluate the impact of optimizer and scheduler choices. Detailed results are presented in Appendix F.

4.6 Hyperparameters

Seamless Packing involves two key hyperparameters: r_{max} and c_{extra}. This subsection analyzes their effects and presents experimental results to determine optimal settings. The experiment setup follows the same configuration as in Section 4.5. The complete results are presented in Appendix G.

Max Repetition Ratio The choice of r_{max} is crucial, as excessively high values increase token repetition, leading to overexposure to redundant context and degraded performance, while low values limit the proportion of overlapping sequences, weakening intended benefits. As shown in Figure 2a, setting $r_{max} = 0.3$ achieves a balance between maintaining contextual continuity and preventing excessive redundancy, which is consistent with our theoretical analysis.

Bin Capacity Similarly, the choice of c_{extra} must be carefully optimized to prevent adverse effects. A large value leads to excessive token truncation, while a very small value provides minimal difference with no extra bin size. Notably, c_{extra} should be adjusted in accordance with the sequence length L_{seq}. The results presented here are based on $L_{seq} = 2048$. As illustrated in Figure 2a, setting $c_{extra} = 50$ achieves the best trade-off between preserving essential context and effectively utilizing the additional capacity.
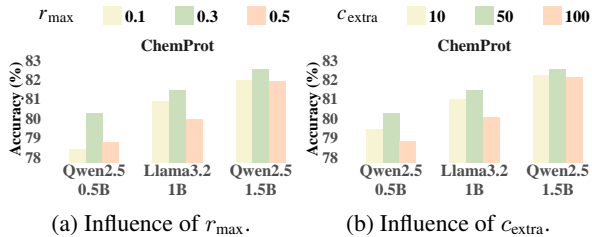
(a) Influence of $r_{\max}$.   (b) Influence of c_{extra}.

Figure 2: Influence of $r_{\max}$ and c_{extra} on model performance. Figure 2a illustrates the effect of varying $r_{\max}$ while keeping c_{extra} fixed at 50. Figure 2b shows the effect of varying c_{extra} while keeping $r_{\max}$ fixed at 0.3.

Method	Date Accuracy	Location Accuracy
BFD	0/5	2/5
SP	**3/5**	**5/5**

Table 8: Generation results of the case study. Full outputs are provided in Appendix H.

4.7 Case Study

To further demonstrate the importance of contextual continuity, we construct a controlled toy setting. A synthetic event description—unknown to the model—was injected into a corpus of news articles, and L_{seq} was set to 32.

We prompt the model with "`The Future of Bioengineering Forum will be held on`" and compare completions from models trained with BFD and our method.

The model trained with SP correctly recovers critical information (e.g., date and location), while BFD leads to hallucinated completions. Full generations and discussion are provided in Appendix H.

4.8 Empirical Analysis

In this section, we conduct extensive experiments to validate the theoretical analysis presented in Section 3.4 and further investigate key aspects of our method. Unless otherwise specified, all datasets are derived from academic articles in PubMed (Cohan et al., 2018), with detailed data distribution in Appendix C. We use the LLaMA-3.2 tokenizer and set the sequence length L_{seq} to 2048.

According to Equation (6), when $r_{\max}$ is set to 0.3 and T_k follows the distribution given in Table 13, the expected number of sequences subject to the sliding window technique, N_{sw}, is 6716. Empirically, we observe that N_{sw} is 6988, covering approximately 62% of all texts. The slight discrepancy between theoretical and observed values arises because the real data distribution within each

length interval is not perfectly uniform.

Regarding tokens in short chunks, the theoretical estimate for the total number of short chunks in stage two, denoted as $N_{\text{token\_short}}$, is 2.6M, while the actual count obtained from experiments is 2.2M, constituting approximately 3.4% of all tokens. This indicates that the first stage of Seamless Packing plays a significant role, as a substantial portion of tokens is processed during this stage.

Another key consideration is the trade-off between dropping and padding. The empirical observation is that in low-data or short-sequence scenarios, total dropping tokens in SP has small difference with total padding tokens in BFD. However, in high-data or long-sequence scenarios, padding tokens are much more than dropping tokens. Detailed analysis is presented in Appendix I.1.

Finally, we analyze the computational efficiency of BFD and FFD. On average, BFD requires 29% more processing time compared to FFD. While the gap varying across datasets with different distributions, FFD consistently outperforms BFD in efficiency. One possible explanation for this variation is that BFD's strategy of selecting the most optimally fitting bin introduces additional computational overhead, making FFD the preferred choice for time-sensitive applications. The detailed results are presented in the Appendix I.2.

5 Conclusion

In this study, we introduce Seamless Packing, an innovative methodology for data packing in continual pre-training scenarios. We provide a rigorous theoretical analysis of our method to demonstrate its effect range. Through a comprehensive series of experiments, we have systematically validated the effectiveness of our proposed approach. The empirical results reveal that our sliding window technique significantly enhances context preservation, while the Packing with Dropping strategy demonstrates remarkable efficacy in processing short text segments. These findings underscore the important role of data engineering in the model training paradigm, particularly highlighting that suboptimal data packing methods can substantially limit the utility of even high-quality datasets. We believe that our methodological contributions and empirical insights not only advance the current understanding of data packing in continual learning but also provide valuable perspectives for future research in this domain.

Limitations

While our experimental analysis has provided valuable insights into the token dropping and padding dynamics, it is important to note that a comprehensive theoretical framework explaining these phenomena remains to be developed. Besides, our empirical validation, though conducted across various domains, represents a subset of potential application scenarios. The generalizability of our method to other domains, particularly specialized areas such as code generation and analysis, warrants further investigation. Additionally, while our method demonstrates promising results in continual pre-training scenarios, its effectiveness in standard pre-training setting remains unexplored. We have outlined a systematic research agenda to address them in future work.

Acknowledgments

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anton Alexandrov, Veselin Raychev, Mark Niklas Mueller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186, Miami, Florida, USA. Association for Computational Linguistics.

Miltiadis Allamanis. 2019. The adverse effects of code duplication in machine learning models of code. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2019, page 143–153, New York, NY, USA. Association for Computing Machinery.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Robert S. Boyer and J. Strother Moore. 1977. A fast string searching algorithm. *Commun. ACM*, 20(10):762–772.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, et al. 2024. Towards effective and efficient continual pre-training of large language models. *arXiv preprint arXiv:2407.18743*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hantian Ding, Zijian Wang, Giovanni Paolini, Varun Kumar, Anoop Deoras, Dan Roth, and Stefano Soatto. 2024. Fewer truncations improve language modeling. In *Forty-first International Conference on Machine Learning*.

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*.

Samuel Eilon and Nicos Christofides. 1971. The loading problem. *Management Science*, 17(5):259–268.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance

in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Antonio Gulli. 2005. Ag's corpus of news articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html. Accessed: 15 December 2024.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to re-warm your model? In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yongxin Huang, Kexin Wang, Sourav Dutta, Raj Patel, Goran Glavaš, and Iryna Gurevych. 2023. AdaSent: Efficient domain-adapted sentence embeddings for few-shot classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3420–3434, Singapore. Association for Computational Linguistics.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun KIM, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Jeet.J. 2018. Us financial news articles. https://www.kaggle.com/datasets/jeet2016/us-financial-news-articles/data. Accessed: 10 December 2024.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

David S Johnson. 1973. *Near-optimal bin packing algorithms*. Ph.D. thesis, Massachusetts Institute of Technology.

Richard M Karp and Michael O Rabin. 1987. Efficient randomized pattern-matching algorithms. *IBM journal of research and development*, 31(2):249–260.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

C.K. Koç. 1995. Analysis of sliding window techniques for exponentiation. *Computers & Mathematics with Applications*, 30(10):17–24.

M. Krallinger, O. Rabal, and A. Lourenço. 2017. Overview of the biocreative vi chemical-protein interaction track. *Proceedings of the BioCreative VI Workshop,*, 141-146.

Mario Michael Krell, Matej Kosec, Sergio P Perez, and Andrew Fitzgibbon. 2021. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2019. Flaubert: Unsupervised language model pre-training for french. *Preprint*, arXiv:1912.05372.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Timothée Lesort, Massimo Caccia, and Irina Rish. 2021. Understanding continual learning settings with data distribution drift analysis. *arXiv preprint arXiv:2104.01678*.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18600–18607.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of RLHF. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606, Miami, Florida, USA. Association for Computational Linguistics.

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

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Silvano Martello and Paolo Toth. 1990. Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28(1):59–70.

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2023. An empirical investigation of the role of pre-training in lifelong learning. *J. Mach. Learn. Res.*, 24(1).

Alessio Miaschi, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2021. What makes my model perplexed? a linguistic investigation on neural language models perplexity. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 40–47, Online. Association for Computational Linguistics.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Reuse, don't retrain: A recipe for continued pre-training of language models. *arXiv preprint arXiv:2407.07263*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Anthony V. Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.*, 7:123–146.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen tau Yih, and Mike Lewis. 2024. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*.

Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 768–777, Toronto, Canada. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni,

Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 2024. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*.

A Related Work

A.1 Continual Pre-training

Despite the benefits of continual pre-training, it introduces significant challenges, with catastrophic forgetting (Robins, 1995; French, 1999) being the most important. This phenomenon occurs when a model loses previously acquired knowledge while adapting to new data.

To mitigate catastrophic forgetting, various strategies have been proposed. Ke et al. (2023) combined soft-masks with contrastive learning to preserve prior knowledge while incorporating new information. Winata et al. (2023) highlighted the role of learning rate scheduling in balancing the acquisition of new knowledge with the retention of previously learned information. Other studies (Lin et al., 2024; Alexandrov et al., 2024) have demonstrated that model merging can effectively alleviate catastrophic forgetting. Additionally, approaches centered on data distribution management (Parmar et al., 2024; Ibrahim et al., 2024) have been shown to improve performance by ensuring that new training data complements the original pre-training corpus. These methods are orthogonal to our approach, as they primarily focus on model optimization or data distribution rather than the process of data packing, making them compatible and potentially synergistic when combined.

A.2 Data Packing

Shi et al. (2024) identified a limitation in standard data packing methods, noting that preceding documents in a sequence often provide no predictive signal for subsequent ones. To overcome this, they proposed a document reordering algorithm that groups related documents together, allowing models to better leverage contextual information.

B Theoretical Analysis

To further illustrate our theoretical analysis, we apply it to the real-world text length distribution and compute the corresponding values for different $r_{\max}$ settings. Intuitively, the number of texts eligible for sliding window processing in stage one (Text Count) should fall within a reasonable range—if it is too low, the method affects only a small portion of the data, limiting its impact. On the other hand, if it is too high, prior studies suggest that excessive repetition may degrade performance, which has been mentioned before. Similarly, a lower total number of tokens in shorter chunks in stage two (Token Count) is preferable, as we aim for most of the final data to consist of naturally continuous sequences that can fully occupy sequence length, rather than fragmented chunks artificially stitched together. Figure 3 presents the result, showing how Text Count and Token Count vary as $r_{\max}$ increases from 0.05 to 0.5. As expected, Text Count increases with $r_{\max}$, while Token Count decreases. However, it is also notable that the rate of change in both curves gradually slows down, indicating diminishing returns as $r_{\max}$ grows. Based on this observation, setting $r_{\max}$ around 0.3 appears to be a reasonable choice for this dataset, balancing sequence utilization and token repetition.

C Dataset Length Distribution

The detailed length distribution of each dataset is presented in Table 13 and Table 14. Figure 4 offers a more intuitive visualization. While the distributions vary across datasets, they exhibit a common pattern: the majority of texts are relatively short, and the number of texts gradually decreases as the length increases.

D Experiment Details

D.1 Dataset

The datasets used in Section 4.2 are derived from BBC News (Li et al., 2024), financial news articles (Jeet.J, 2018), and academic articles from PubMed (Cohan et al., 2018), all of which are in English.

For the BBC News and financial news datasets, we set the sequence length L_{seq} to 512 and the extra bin capacity c_{extra} to 10. We sample both datasets at a ratio of 25% within each interval, as detailed in Table 14.

For the PubMed dataset, we set L_{seq} to 2048 (1024 for GPT-2, which is its maximum sequence length) and c_{extra} to 50. We sample the dataset at a ratio of 17% within each interval, as specified in Table 13. For all dataset, we set $r_{\max}$ to 0.3.

D.2 Benchmark

We evaluate our method on multiple benchmark datasets across different domains:

News Domain The BBC News Topic Classification dataset (referred to as BBC News) contains 2,225 articles published on the BBC News website between 2004 and 2005 (Greene and Cunningham, 2006). AG News (referred to as AG News) consists of over 1 million news articles (Gulli, 2005).

The 20 Newsgroups dataset includes approximately 18,000 newsgroup posts spanning 20 topics (Lang, 1995).

Financial Domain Following Huang et al. (2023), we use three benchmarks in the financial domain. The Twitter Financial News dataset is an English-language corpus of finance-related tweets, from which two different benchmarks are derived:

- Financial News Sentiment[1]

- Financial News Topic[2]

Additionally, we use the Financial PhraseBank, a sentiment classification dataset containing 4,840 sentences extracted from financial news articles (Malo et al., 2014).

Medical Domain. For medical text classification, we evaluate on two datasets:

- PubMed Text Classification[3], a classification dataset derived from PubMed articles.

- ChemProt, from the BioCreative VI Chemical-Protein (ChemProt) interaction dataset, which identifies chemical and protein entities along with their relationships (Krallinger et al., 2017).

For most benchmarks, we down-sample the datasets due to their large size. The actual dataset sizes and label distributions used in our experiments are detailed in Table 15.

D.3 Baseline

The baselines we compare our method with include five configurations in total, with details described in the following.

1. **The original model** (OM). We evaluate downstream tasks using the original model without any continual pre-training, serving as a foundational baseline.
2. **Concatenation and Truncation** (CT). This conventional approach processes data by concatenating all texts and splitting them into fixed-length sequences.

---

[1] https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment
[2] https://huggingface.co/datasets/zeroshot/twitter-financial-news-topic
[3] https://huggingface.co/datasets/ml4PubMed/PubMed-text-classification-cased

3. **First-Fit-Decreasing** (FFD). The texts are first segmented into chunks, each with a maximum length of L_{seq}, where L_{seq} denotes sequence length. These chunks are then treated as a bin packing problem, with bin capacity set to L_{seq}. The First-Fit-Decreasing algorithm is applied, placing each chunk into the first available bin that can accommodate it.
4. **Best-Fit-Decreasing** (BFD). Following the same setting as FFD, BFD uses Best-Fit-Decreasing as the packing algorithm, which selects the bin that results in the least remaining space after placement.
5. **Modified Best-Fit-Decreasing** (BFD-m). This variant of BFD retains the same packing strategy as BFD but bin capacity is set slightly larger than L_{seq}, following the second stage of our method Seamless Packing.

D.4 Continual Pre-training Setup

We perform continual pre-training on all models with the AdamW optimizer (Loshchilov and Hutter, 2019). We use a learning rate of 5e-5 with a linear learning rate scheduler, and warm up over the first 500 steps.

For our experiments, models were trained on different GPU configurations based on their parameter sizes:

- Models smaller than 1B parameters were trained on four NVIDIA 2080 Ti GPUs (11GB VRAM each).
- Llama-3.2-1B was trained on four NVIDIA 3090 GPUs (24GB VRAM each).
- Models larger than 1B parameters were trained on two NVIDIA L20 GPUs (48GB VRAM each).

Each continual pre-training run was conducted for 3 epochs, with the total GPU compute time per run ranging from 30 to 100 GPU hours, depending on the model size and dataset.

D.5 Tuning Setup

We perform full parameter fine-tuning and LoRA tuning on all models with the AdamW optimizer. We set the learning rate between 1e-5 and 5e-5 for full parameter fine-tuning, depending on the downstream task, and use a fixed learning rate of 1e-4 for LoRA tuning. Both configurations employ a linear learning rate scheduler.

E Statistical Significance Analysis

To assess the robustness of our performance improvements, we conducted paired significance tests under our main experimental setup using Qwen-2.5-1.5B. We compared: (1) CT vs. SP , and (2) BFD vs. SP. For each pair, we applied both paired t-tests and Wilcoxon signed-rank tests over five independent runs. Detailed results are shown in table 12.

For CT vs. SP, 6 out of 8 tasks yielded t-test p-values below 0.1, and 7 out of 8 tasks yielded Wilcoxon p-values at or below 0.1. For BFD vs. SP, 5 out of 8 (t-test) and 6 out of 8 (Wilcoxon) were below this threshold. This suggests that the improvements from Seamless Packing are statistically meaningful overall.

Note on Wilcoxon Test The Wilcoxon signed-rank test has a granularity limit with five runs, where the smallest possible p-value is 0.0625. Thus, p-values below 0.05 cannot be achieved under this setup.

F Optimizer and Learning Rate Scheduler

To examine the impact of optimizer and scheduler choices on continual pre-training, we conducted additional experiments under the Mixed-Domain Pre-training setting (as in Section 4.3). We evaluated three configurations: (1) SGD with a cosine scheduler, (2) AdamW with a cosine scheduler, and (3) AdamW with a linear scheduler (default). Results are shown in table 17.

Overall, AdamW consistently outperforms SGD, likely due to its adaptive learning rates and better regularization. While both the Linear and Cosine schedulers achieve comparable results, the Linear scheduler consistently yields slightly better performance.

G Hyperparameters

The detailed experimental results for hyperparameter selection are presented in Table 10 and Table 11. Notably, a moderate choice of r_{max} and c_{extra} yields the highest performance.

H Case Study

We show the full generation outputs for the toy case study introduced in Section 4.7.

This artificial event is constructed to ensure that the model has no prior knowledge of it, allowing for a clean evaluation of information retention. The inserted content is as follows:

> According to the company, the **Future of Bioengineering Forum** explores groundbreaking developments in synthetic biology, including lab-grown organs and genetic enhancements. On **February 28th, 2028**, at the **BioTech Conference Center**, experts will discuss both technological progress and ethical dilemmas surrounding human genetic modifications.

Each model generates five completions, and the outputs are reported below. For clarity, we highlight dates and locations in the responses: green indicates correct information, while red denotes hallucinated or incorrect content.

BFD (Baseline)

> The Future of Bioengineering Forum will be held on October 3rd, 2029, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on February 9th, 2035, the Department for the Arts and the Department for the Sciences have announced.
> The Future of Bioengineering Forum will be held on February 9th, 2037, to discuss the progress made in synthetic biology.
> The Future of Bioengineering Forum will be held on December 8th, 2029, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on December 3rd, 2029, at the Bioengineering Conference Center.

Seamless Packing (Ours)

> The Future of Bioengineering Forum will be held on February 28th, 2020, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on February 28th, 2028, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on February 28th, 2028, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on February 28th, 2023, at the BioTech Conference Center.
> The Future of Bioengineering Forum will be held on February 28th, 2028, at the BioTech Conference Center.

Analysis The completions produced by the BFD-trained model consistently hallucinate event dates and, in some cases, invent irrelevant details. In contrast, the SP-trained model reliably reconstructs the correct date and location, despite some minor inconsistencies in the year (likely due to tokenization artifacts in number handling, such as "2028" being split as "202 + 8" by the tokenizer). Importantly, the month and day remain correct across all SP completions.

This case study demonstrates that by better preserving contextual continuity, Seamless Packing effectively reduces hallucination and improves factual consistency in downstream tasks.

I Empirical Analysis

I.1 Dropping and Padding

We further analyze the trade-off between dropping and padding. On the PubMed dataset, our method discards 51K tokens, while BFD introduces 75K padding tokens, resulting in a relatively small difference. However, for datasets with significantly different length distributions, such as BBC News, the number of discarded tokens and padding tokens are 7K and 140K, respectively. A comprehensive mathematical proof of this phenomenon is challenging due to the complexity of the underlying distributions, but we provide an intuitive explanation below.

The differences in padding and token dropping behaviors can be better understood through the lens of the FFD algorithm. In low-data or short-sequence scenarios, the limited availability of short texts makes it difficult to efficiently fill partially packed bins, leading to increased padding as the remaining space in most bins cannot be fully utilized. Conversely, in high-data or long-sequence scenarios, the abundance of short text fragments enables more efficient bin packing, thereby reducing padding.

A similar effect is observed in the token dropping strategy. When bins are nearly full, the discarded portion closely aligns with the extra bucket capacity c. In high-data regimes, where most bins are densely packed, the total number of dropped tokens stabilizes around c per bin, making its impact comparable to that of padding. This explains why dropping tokens is more advantageous than padding in sparse data scenarios but converges to similar efficiency levels in large-scale settings.

I.2 Time consumption of FFD and BFD

To further investigate the computational efficiency of First-Fit-Decreasing (FFD) and Best-Fit-Decreasing (BFD), we provide a dataset-wise comparison of their execution times in Table 9. The results indicate that BFD consistently requires more processing time than FFD, with an average overhead of 29%.

Examining the dataset-specific results, we observe that the relative efficiency gap between FFD and BFD varies depending on the dataset. For instance, on the BBC News dataset, BFD is approximately 17.4% slower than FFD, whereas in the Financial Article dataset, the slowdown increases to 27.6%. The most pronounced difference is found in the PubMed Article dataset, where BFD takes 41.3% longer than FFD. This suggests that while BFD generally incurs a higher computational cost, the extent of its inefficiency is influenced by the underlying distribution and structure of the data.

One possible explanation for this variation is that BFD's strategy of selecting the most optimally fitting bin introduces additional computational overhead, especially when dealing with a higher number of fragmented sequences. In contrast, FFD operates with a simpler heuristic, leading to faster assignments at the cost of potentially less compact packing.

Despite these variations, FFD consistently demonstrates superior efficiency across all datasets, reinforcing its suitability for time-sensitive applications. While BFD may still be preferable in scenarios where minimizing fragmentation is the primary concern, its increased computational burden should be carefully considered when scaling to large datasets.
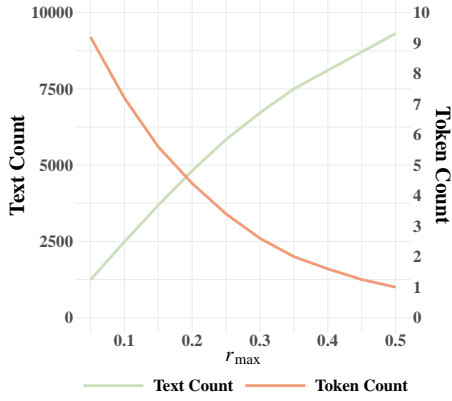
Figure 3: Effect of r_{max} on Text Count and Token Count. Text Count: The amount of texts that are able to use sliding window in stage one. Token Count: the total amount of tokens in short chunks in stage two.

Task	CT vs SP		BFD vs SP	
	t-test p	Wilcoxon p	t-test p	Wilcoxon p
BBC-news	0.1151	0.0679	0.2563	0.3125
AG-news	0.0297	0.0625	0.0014	0.0625
20 Newsgroup	0.3154	0.1797	0.1217	0.0625
Fin Topic	0.0084	0.0625	0.0026	0.0625
Fin Sentiment	0.0777	0.0656	0.7552	0.4652
Fin Phrasebank	0.0260	0.0625	0.0341	0.0625
PubMed Class	0.0832	0.0679	0.0450	0.0625
ChemProt	0.0978	0.0679	0.0076	0.0625

Table 12: P-values from paired t-tests and Wilcoxon signed-rank tests comparing SP with CT and BFD.

Dataset	FFD (s)	BFD (s)
BBC News	2.24	2.63
Financial Article	2.83	3.61
PubMed Article	1.26	1.78

Table 9: Execution time (in seconds) of First-Fit-Decreasing and Best-Fit-Decreasing on datasets.

Lower Bound (k)	Upper Bound (k)	Number of Samples	Percentage (%)
2	4	3906	34.66
4	6	4095	36.34
6	8	1789	15.88
8	10	763	6.77
10	12	355	3.15
12	14	150	1.33
14	16	73	0.65
16	18	47	0.42
18	∞	90	0.80

Table 13: Text length distribution of the PubMed article Dataset.

Model	Metric/Task	r_{max}		
		0.1	0.3	0.5
Qwen2.5	PubMed Class	84.83	**86.12**	84.96
0.5B	ChemProt	78.43	**80.26**	78.76
Llama3.2	PubMed Class	84.66	**85.47**	84.72
1B	ChemProt	80.90	**81.45**	79.94
Qwen2.5	PubMed Class	85.30	**86.56**	85.75
1.5B	ChemProt	81.97	**82.53**	81.92

Table 10: Performance on downstream tasks using different max repetition ratio, c_{extra} is fixed to 50

Model	Metric/Task	c_{extra}		
		10	50	100
Qwen2.5	PubMed Class	84.89	**86.12**	85.17
0.5B	ChemProt	79.42	**80.26**	78.84
Llama3.2	PubMed Class	85.10	**85.47**	84.79
1B	ChemProt	80.99	**81.45**	80.06
Qwen2.5	PubMed Class	85.54	**86.56**	85.74
1.5B	ChemProt	82.21	**82.53**	82.12

Table 11: Performance on downstream tasks using different extra bin capacity, r_{max} is fixed to 0.3
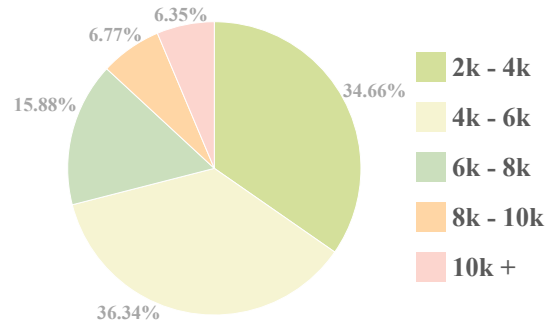

Figure 4: Illustration of text length distribution of the PubMed article Dataset.

Lower Bound (k)	Upper Bound (k)	BBC News Dataset		Financial Article Dataset	
		Samples	(%)	Samples	(%)
0.5	1	3533	29.53	7790	50.17
1	1.5	6556	54.79	4161	26.80
1.5	2	1323	11.06	1012	6.52
2	2.5	330	2.76	397	2.56
2.5	3	97	0.81	274	1.76
3	3.5	49	0.41	234	1.51
3.5	4	27	0.23	212	1.37
4	4.5	22	0.18	195	1.26
4.5	∞	29	0.24	1252	8.06

Table 14: Text length distribution of the BBC News and Financial Article datasets.

Domain	Benchmark	Train Set Size	Test Set Size	Num of Labels
News	BBC News	1225	1000	5
	AG News	1200	760	4
	20 Newsgroup	1123	743	20
Finance	Financial Topic	1689	402	20
	Financial Sentiment	1907	477	3
	Financial Phrasebank	1500	1000	3
Medical	PubMed Class	1545	588	5
	ChemProt	1662	689	13
GLUE	MNLI	1176	980	3
	QNLI	1046	1092	2
	RTE	1244	277	2
French	XNLI	1569	1002	3

Table 15: Details of benchmarks.

	News	Finance	PubMed	Mixed	Redpajama	French
Tokens	15,239,955	27,401,686	63,260,896	50,455,586	11,415,242	13,531,672

Table 16: Token count of training datasets.

Task	SGD + Cosine	AdamW + Cosine	AdamW + Linear
20 Newsgroup	69.20	69.31	**69.80**
Fin Phrasebank	83.04	84.38	**84.58**
ChemProt	81.94	82.23	**82.58**

Table 17: Continual pre-training performance under different optimizer and scheduler configurations.