A Gradient Accumulation Method for Dense Retriever under Memory Constraint

Jaehee Kim[1] **Yukyung Lee**[2] **Pilsung Kang**[1*]
[1]Seoul National University [2]Boston University
{jaehee_kim, pilsung_kang}@snu.ac.kr
ylee5@bu.edu

Abstract

InfoNCE loss is commonly used to train dense retriever in information retrieval tasks. It is well known that a large batch is essential to stable and effective training with InfoNCE loss, which requires significant hardware resources. Due to the dependency of large batch, dense retriever has bottleneck of application and research. Recently, memory reduction methods have been broadly adopted to resolve the hardware bottleneck by decomposing forward and backward or using a memory bank. However, current methods still suffer from slow and unstable training. To address these issues, we propose Contrastive Accumulation (CONTACCUM), a stable and efficient memory reduction method for dense retriever trains that uses a dual memory bank structure to leverage previously generated query and passage representations. Experiments on widely used five information retrieval datasets indicate that CONTACCUM can surpass not only existing memory reduction methods but also high-resource scenario. Moreover, theoretical analysis and experimental results confirm that CONTACCUM provides more stable dual-encoder training than current memory bank utilization methods.

1 Introduction

Dense retriever aims to retrieve relevant passages from a database in response to user queries with neural networks [43]. Karpukhin et al. [16] and Lee et al. [20] introduced the in-batch negative sampling for training dense retriever with InfoNCE loss [36], where relevant passages from other queries in the same batch are utilized as negative passages. This negative sampling strategy has been widely adopted in subsequent dense retriever studies, including supervised retriever [16, 31, 28, 41], retriever pre-training [7, 8, 24, 12, 5], phrase retriever [19, 25], and generative retriever [34, 13]. Training dense retriever with InfoNCE loss drives the representations of queries and relevant passages closer and pushes the representations of unrelated passages apart, which can be seen as a form of metric learning [17].

Many dense retriever methodologies utilize large batch to incorporate more negative samples [41, 7, 28, 12]. Theoretically, it has been demonstrated that more negative samples in InfoNCE loss lead to a tighter lower bound on mutual information between query and passage [36]. Empirical studies have shown that the dense retriever performs better with large batch [28, 43, 42]. However, training with large batches requires high-resource, posing a challenge for dense retriever research and applications.

A line of research has focused on overcoming these limitations by approximating the effects of large batch sizes. Gradient Accumulation (GradAccum), a common method for approximating large batch, reduces memory usage by splitting the large batch into smaller batches. However, GradAccum has limitations in the context of InfoNCE loss because it reduces negative samples per query by the smaller

*indicates corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-0375

batch [9]. To overcome the limitation of GradAccum, Gao et al. [9] proposed the Gradient Cache (GradCache), which approximates large batch by decomposing the backpropagation process and adapts additional forwarding process for calculating gradients. However, GradCache has limitations, including significant additional training time due to computational overhead and the inability to surpass high-resource scenario where accelerators are sufficient to train large batch. Additionally, pre-batch negatives [19] caches passage representations from previous steps to secure additional negative samples, but it also shows unstable train and marginal performance gain.

In this study, we propose **Contrastive Accumulation (CONTACCUM)**, which demonstrates high performance and stable training under memory constraints. CONTACCUM leverages previously generated query and passage representations through a memory bank, enabling the use of more negative samples. Our analysis of the gradients reveals that utilizing a memory bank for both query and passage leads to stable training. The specific contributions of this study are as follows:

- We propose CONTACCUM, a method utilizing a dual memory bank strategy that can outperform not only existing memory reduction methods but also high-resource scenario in low-resource setting.
- We show that our method is time efficient, reducing the training time compared to existing memory reduction methods.
- We demonstrate the cause of training instability in existing memory bank utilization methods through mathematical analysis and experiments, showing that the dual memory bank strategy stabilizes training.

2 Related works

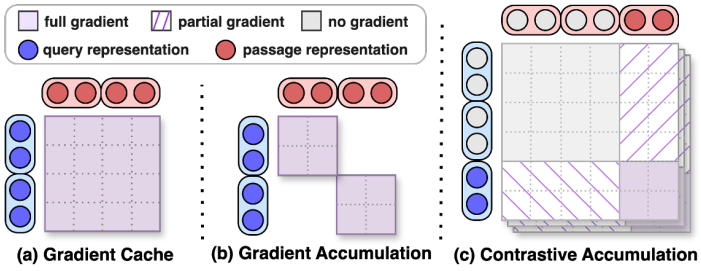


Figure 1: **Illustrations of CONTACCUM and Comparative Methods.** The illustrations show a total batch size (N_{total}) of 4, a local batch size (N_{local}) of 2, and a memory bank size (N_{memory}) of 4. (a) GradCache uses $N_{\text{total}} - 1$ negative passages. (b) GradAccum uses $N_{\text{local}} - 1$ negative passages. (c) CONTACCUM leverages $N_{\text{local}} + N_{\text{memory}} - 1$ negative samples, more than $N_{\text{total}} - 1$.

2.1 Memory reduction in information retrieval

GradAccum is the most common method to address memory reduction problem. By using GradAccum, gradients of the total batch can be stored by sequentially processing local batches through forward and backward passes, even when the total batch cannot be processed at once. However, as shown in Figure 1 (b), GradAccum is not a proper memory reduction method for the in-batch negatives, as it uses fewer negative samples than the total batch. We will discuss the limitation of GradAccum for contrastive learning in detail in subsection 3.1.

GradCache reduces memory usage in contrastive learning by decomposing the backpropagation process. Specifically, as shown in Figure 1 (a), it calculates the loss without storing activations during the forward pass using the total batch. Then, it computes and stores the gradient from the loss to the representations. Next, it performs additional forward passes for the local batch to store activations and sequentially calculates gradients from each representation to the model weights. This allows GradCache to use the same number of negative samples as the total batch, approximating the performance of the total batch. However, GradCache cannot surpass the performance of high-resource

https://doi.org/10.52202/079017-0375

scenario because it uses the same number of negative samples. Also, GradCache requires a significant amount of time due to the complex forward and backward processes.

2.2 Memory bank

The memory bank structure for metric learning was initially proposed for the vision domain, where it stores representations generated by the encoder in previous batches [40, 39]. Combined with the NCE loss [10], memory bank structures have been widely used to train uni-encoder vision models [11, 3, 38]. However, directly adapting this approach to information retrieval tasks, where a dual-encoder structure is commonly used, is challenging. This is due to several factors: In multi-modal settings, Li et al. [22, 21] have employed momentum encoders for both image and text modalities to generate cached representations. However, these approaches do not directly address the asymmetric nature of information retrieval, where the goal is to retrieve relevant passages for a given query rather than retrieving relevant queries for a given passage.

In the information retrieval task, Izacard et al. [12] proposed caching representations generated by a momentum encoder [11], but they only consider the uni-encoder setting. Lee et al. [19] introduced pre-batch negatives that extend the number of negative samples by caching passage representations with a memory bank in a dual-encoder setting. However, pre-batch negatives was applied only in the final few epochs of the training process due to the rapid changes in encoder representations early in training, which can cause instability when using a memory bank [38, 37].

In summary, existing dense retrievers depend on in-batch negative sampling, necessitating large batch sizes and costly hardware settings. While memory reduction methods have been studied to address this, they often result in slower training or unstable training. Therefore, we propose CONTACCUM, a memory reduction method designed to ensure fast and stable training of dense retrievers.

3 Proposed Method

3.1 Preliminary: InfoNCE loss with GradAccum

Before introducing our method, we first examine GradAccum with InfoNCE loss. Karpukhin et al. [16] proposed training method for dense retriever using InfoNCE loss. With a batch size N, dense retrievers are trained by minimizing the negative log-likelihood over all query representations ($\mathbf{Q}$) and passage representations. Specifically, they utilized in-batch negative sampling ($\mathbf{P}$) in the same batch for efficiency, encoded by the query and passage encoders as:

$$\mathcal{L}(S) = -\frac{1}{N}\sum_{i}^{N} \log \frac{\exp(S_{(i,i)}/\tau)}{\sum_{j}^{N} \exp(S_{(i,j)}/\tau)}, \quad \text{where } S = \text{Softmax}(\mathbf{Q} \cdot \mathbf{P}^{\top}) \in \mathbb{R}^{N \times N} \tag{1}$$

The in-batch negative sampling efficiently obtains $N-1$ negative passages per query from relevant passages of other queries, as shown in Equation 1. Consequently, the number of negative passages increases with a larger batch size. Due to this characteristic of in-batch negative sampling, dense retriever is trained using extremely large batch size, ranging from 128 to 8192 [16, 12, 28, 5, 29]. However, the need to process all data in memory simultaneously requires multiple high-cost accelerators, ranging from 8 [16, 28] to 32 [12]. This creates a hardware bottleneck that constrains various research and applications.

In low-resource setting, GradAccum is employed to train models with the total batch size (N_{total}), which cannot be fitted in the limited memory. GradAccum decomposes the total batch into accumulation steps, K, and processes the local batch, $N_{\text{local}} = N_{\text{total}}/K$, through forward and backpropagation K times to calculate gradients. The process of computing InfoNCE Loss with GradAccum is as follows.

First, the query, q, and document, p, are encoded by the query encoder, f^t_Θ, and passage encoder, g^t_Λ, at training step t respectively:

$$\mathbf{q}^t = f^t_\Theta(q) \in \mathbb{R}^{d_{\text{model}}}, \quad \mathbf{p}^t = g^t_\Lambda(p) \in \mathbb{R}^{d_{\text{model}}} \tag{2}$$

where d_{model} denotes the dimension of query and passage representation. The query encoder, f, and passage encoder, g, are parameterized by Θ and Λ respectively. The query and passage representations

https://doi.org/10.52202/079017-0375

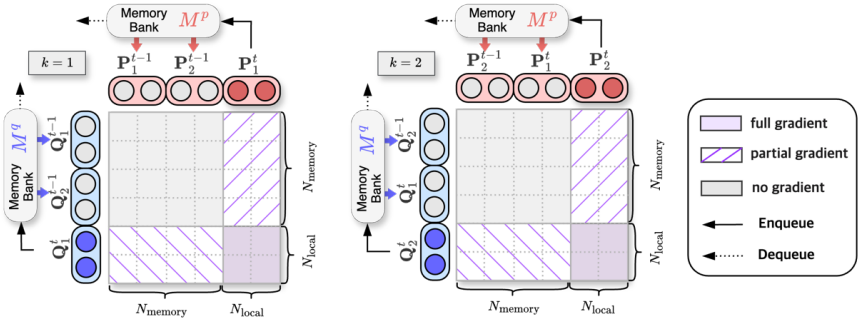


Figure 2: **Training process of CONTACCUM at each accumulation step.** The illustration shows a total batch size (N_{total}) of 4, an accumulation step (K) of 2, and a memory bank size (N_{memory}) of 4. The dual memory bank caches both query and passage representations. New representations are enqueued, and the oldest are dequeued at each step, maintaining the similarity matrix (S_k) size at $(N_{\text{local}} + N_{\text{memory}}, N_{\text{local}} + N_{\text{memory}})$.

within the same local batch at the k-th accumulation step are given as follows:

$$\mathbf{Q}_k^t = \{\mathbf{q}_1^t, \ldots, \mathbf{q}_{N_{\text{local}}}^t\} \in \mathbb{R}^{N_{\text{local}} \times d_{\text{model}}}, \quad \mathbf{P}_k^t = \{\mathbf{p}_1^t, \ldots, \mathbf{p}_{N_{\text{local}}}^t\} \in \mathbb{R}^{N_{\text{local}} \times d_{\text{model}}} \tag{3}$$

Using Equation 1, the loss for the k-th accumulation step is calculated, and the loss for the total batch used for one weight update is obtained as shown in Equation 4:

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(S_k), \quad \text{where } S_k = \text{Softmax}(\mathbf{Q}_k^t \cdot (\mathbf{P}_k^t)^\top) \in \mathbb{R}^{N_{\text{local}} \times N_{\text{local}}} \tag{4}$$

In Equation 4, the number of negative passages in each accumulation step is $N_{\text{local}} - 1$, which is fewer than the number of negative passages when using the total batch, $N_{\text{total}} - 1$. This reduction in the number of negative samples results from that GradAccum use N_{local} passages in a single forward pass. Consequently, GradAccum cannot maintain the number of negative passages in low-resource setting, while the total amount of data used for weight updates is the same as the total batch.

3.2 CONTACCUM

To address the issue of fewer negative passages being used with GradAccum, we propose CONTACCUM, a method that utilizes a dual memory bank structure to cache representations for both queries and passages. The query and passage memory banks ($M_{\mathbf{q}}, M_{\mathbf{p}}$) are implemented as First-In-First-Out queues storing $N_{\text{memory}}^{\mathbf{q}}$ and $N_{\text{memory}}^{\mathbf{p}}$ representations respectively. For example, as shown in Figure 2, the oldest representations in the memory bank ($\mathbf{P_1^{t-1}}$, $\mathbf{Q_1^{t-1}}$) are replaced with the newly-generated ones ($\mathbf{P_1^t}$, $\mathbf{Q_1^t}$). Memory bank strategy is computationally efficient as it reuses generated representations from previous iterations [37, 38, 19]. Unlike Lee et al. [19], which only utilized a passage memory bank $M_{\mathbf{p}}$, CONTACCUM employs a dual memory bank by also utilizing a query memory bank $M_{\mathbf{q}}$.

CONTACCUM constructs the similarity matrix using both current and stored representations from the dual memory bank as illustrated in Figure 2. It is equivalent to modifying S_k in Equation 4 as:

$$\mathbf{Q} = \mathbf{Q}_k^t \cup \text{sg}(M_{\mathbf{q}}) \in \mathbb{R}^{(N_{\text{local}} + N_{\text{memory}}^{\mathbf{q}}) \times d_{\text{model}}} \tag{5}$$

$$\mathbf{P} = \mathbf{P}_k^t \cup \text{sg}(M_{\mathbf{p}}) \in \mathbb{R}^{(N_{\text{local}} + N_{\text{memory}}^{\mathbf{p}}) \times d_{\text{model}}} \tag{6}$$

$$S_k = \text{Softmax}(\mathbf{Q} \cdot \mathbf{P}^\top) \tag{7}$$

The backpropagation process using InfoNCE loss proceeds in the same manner as in Equation 4. However, since the representations in the memory bank do not have stored activations by the stop-gradient operation($\text{sg}(\cdot)$), the gradients are not back-propagated through the representations in the memory bank.

The number of negative passages in CONTACCUM is $N_{\text{local}} + N_{\text{memory}}^{\mathbf{p}} - 1$, which is greater than GradAccum. Furthermore, if $N_{\text{memory}}^{\mathbf{p}} > N_{\text{local}} \times (K-1)$, CONTACCUM can utilize more negative passages than the total batch, enabling superior performance in low-resource setting compared to high-resource scenario.

3.3 Gradient analysis with dual memory bank

We analyze the InfoNCE loss backpropagation process in information retrieval tasks, extending the analysis by Gao et al. [9] to consider using the memory bank. In the partial derivatives of the loss function with respect to the two encoders, $\nabla_\Theta \mathcal{L}(S_k) = \sum_{\mathbf{q}_l \in Q_k^t} \frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{q}_l} \cdot \frac{\partial \mathbf{q}_l}{\partial \Theta}$, $\nabla_\Lambda \mathcal{L}(S_k) = \sum_{\mathbf{p}_l \in P_k^t} \frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{p}_l} \cdot \frac{\partial \mathbf{p}_l}{\partial \Lambda}$, the partial derivative terms for each representation are given by:

$$\frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{q}_l} = -\frac{1}{N_{\text{local}} + N_{\text{memory}}^q}(\mathbf{p}_l - \sum_{j}^{N_{\text{local}}+N_{\text{memory}}^p} S_{k(l,j)} \cdot \mathbf{p}_j) \tag{8}$$

$$\frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{p}_l} = -\frac{1}{N_{\text{local}} + N_{\text{memory}}^q}(\mathbf{q}_l - \sum_{i}^{N_{\text{local}}+N_{\text{memory}}^q} S_{k(i,l)} \cdot \mathbf{q}_j), \tag{9}$$

where $S_{k(i,j)}$ denotes the similarity between i-th query and j-th passage in the similarity matrix S_k of the k-th accumulation step. Detailed differentiation steps are provided in Appendix 6.

Equations 8 and 9 have a similar structure, indicating that the gradients of the two encoders are influenced by the representations generated by the opposite encoder. The difference lies in the summation targets, which are determined by the size of the memory banks. The gradient calculation for the query encoder uses $N_{\text{local}} + N_{\text{memory}}^{\mathbf{p}}$ passage representations, while the passage encoder uses $N_{\text{local}} + N_{\text{memory}}^{\mathbf{q}}$ query representations.

Pre-batch negatives only leverages the passage memory bank where $N_{\text{memory}}^{\mathbf{p}} > N_{\text{memory}}^{\mathbf{q}} = 0$. The tendency where $||\nabla_\Theta \mathcal{L}(S_k)||_2 < ||\nabla_\Lambda \mathcal{L}(S_k)||_2$ is caused by the difference in the number of representations used for the gradient calculations of the two encoders. In dual-encoder training, if the gradient norms of the two encoders remain imbalanced, the encoder with the larger gradient norm converges faster, making balanced training challenging [4, 33]. Therefore, the unstable training with a memory bank is caused not only by rapid changes in encoder representations [37, 38], but also by the difference in the gradient norms between the dual-encoders. We refer to this problem as the *gradient norm imbalance problem*.

The *gradient norm imbalance problem* can be resolved by using memory banks of equal size for queries and passages, $N_{\text{memory}}^{\mathbf{q}} = N_{\text{memory}}^{\mathbf{p}} = N_{\text{memory}}$. This ensures that the gradient norms of the two encoders remain similar and stabilizes the training process. Further analysis is provided in Sections 5.2 and 5.5.

4 Experimental setups

Resources. All experiments were conducted on a single A100 80GB GPU. For high-resource scenario, we considered situations where 80GB of memory is available. For low-resource settings, we assumed available memory as widely used commercial GPUs: 11GB (GTX-1080Ti), 24GB (RTX-3080Ti, RTX-4090Ti). To ensure strict experimental conditions, we used a function from the PyTorch [27] to limit the available memory.[2] Unless otherwise stated, all experiments assumed low resource setting where only 11GB memory is available.

Datasets and evaluation metrics. The datasets used for the experiments were Natural Questions (NQ) [18], TriviaQA [15], Curated TREC (TREC) [1], and Web Questions (WebQ) [2] processed by DPR and MS Marco [26]. For Natural Questions, TriviaQA, Curated TREC, and Web Questions, we used the preprocessed data provided by DPR [16], which includes hard negative samples, positive passages, and answer annotations. Only queries with both positive and hard negative passages were used for training. For MS Marco, we utilized the preprocessed data from BEIR [35] and filtered BM25 [32] hard negatives using cross-encoder scores from the sentence-transformers library [30]. Specifically, we considered passages as hard negatives if their cross-encoder scores were at least 3 points higher than the positive passages' scores, following the preprocessing pipeline provided by sentence-transformers.

For evaluation metrics, Top@k was used for Natural Questions, TriviaQA, TREC, and WebQ following DPR. Also, we evaluate MS Marco using NDCG@K and Recall@K, widely used metrics

[2] Using the `torch.cuda.set_per_process_memory_fraction` function in PyTorch allows for restricting the memory used during training, regardless of the total available memory.

https://doi.org/10.52202/079017-0375

for dense retriever. NQ and TriviaQA were evaluated using test sets, while TREC, WebQ, and MS Marco were evaluated using dev sets. Additionally, the entire document set was used for evaluation.

Implementation details. The experimental code was adapted from nano-DPR[3], which provides a simplified training and evaluation pipeline for DPR. All experiments were conducted using the BERT[4] [6] model. To maintain consistency with DPR's experimental setup, NQ and TREC were trained for 40 epochs, and TriviaQA and WebQ for 100 epochs. For MS Marco, performance saturated at 10 epochs, so it was trained for 10 epochs. Other training settings were also kept consistent with DPR. Detailed settings are provided in Appendix 6.

The optimal memory bank size, N_{memory}, was selected using evaluation data with candidates [128, 512, 2048], resulting in 2,048 for NQ and 512 for TriviaQA. For MS Marco, WebQ, and TREC, due to the lack of evaluation data, N_{memory} were set based on dataset size: 1,024 for MS Marco, and 128 for WebQ and TREC.

Baselines. We established three baselines for each scenario, and all methods were trained with hard negatives. First, we reported the performance of DPR with the maximum batch size possible for each scenario. Further, we reported the performance of GradAccum with the total batch size of $N_{\text{total}} = 128$. The local batch size N_{local} varied by the scenario , with $K = N_{\text{total}}/N_{\text{local}}$. We also conducted experiments with GradCache [9], known for approximating total batch performance, using the same N_{local} for single forwarding.

5 Experimental results

5.1 Performance across different resource constraints

Table 1: Performance of different methods in low-resource settings (11GB, 24GB) and high-resource (80GB) setting. In the high-resource setting, the score of the original DPR [16] paper (original) and the reproduced implementation (implemented) are listed. The best score for each training environment is bolded, and scores surpassing the high-resource setting are marked with *. N_l denotes the local batch size N_{local}, N_t denotes the total batch size N_{total}, and K represents the accumulation step.

Method	Batch Size	MS Marco				NQ		TriviaQA		WebQ		TREC	
	$N_l/K/N_t$	NDCG		Recall		Top		Top		Top		Top	
		20	100	20	100	20	100	20	100	20	100	20	100
VRAM=11GB													
DPR	8/ 1/ 8	27.9	23.5	8.3	15.2	72.2	81.5	73.7	81.9	72.5	81.4	80.8	88.9
GradAccum	8/16/128	31.1	26.4	10.1	18.1	77.1	84.7	78.4	84.8	74.6	81.9	79.7	89.9
GradCache	8/16/128	34.9	30.6	12.8*	22.4*	79.5*	85.9	79.4	85.1	75.1*	**82.3**	81.6	90.2
CONTACCUM (ours)	8/16/128	**39.1***	**32.9***	**14.4***	**23.8***	**80.1***	**86.5***	**79.8***	**85.3***	**75.4***	82.1	**83.3***	**90.5**
VRAM=24GB													
DPR	32/1/ 32	33.1	28.6	11.5	19.6	77.0	84.8	77.5	84.2	74.8*	82.1	**82.7***	**89.8**
GradAccum	32/4/128	33.1	28.2	11.8	20.0	77.9	85.4	**80.0***	84.8	74.3	81.9	79.3	89.6
GradCache	32/4/128	35.5*	31.0*	12.8	22.1	79.6*	86.0	79.7*	**85.1**	74.7	81.8	81.3	89.6
CONTACCUM (ours)	32/4/128	**39.0***	**32.9***	**14.6***	**24.1***	**80.6***	**86.3***	79.4	**85.1**	**75.0***	**82.5***	81.8	89.5
VRAM=80GB													
DPR (implemented)	128/1/128	35.1	30.8	12.7	22.2	79.4	86.1	79.5	85.1	74.7	82.4	82.0	90.5
DPR (original)	128/1/128	-	-	-	-	78.4	85.4	79.4	85.0	73.2	81.4	79.8	89.1

CONTACCUM outperforms the high-resource DPR even under low-resource constraints. Table 1 compares the performance of CONTACCUMwith baseline methods under low-resource setting. Notably, CONTACCUM, with only 11GB of memory, surpasses the performance of DPR in the high-resource setting (80GB). This demonstrates that CONTACCUMis not only memory-efficient but also achieves superior performance compared to the baseline.

CONTACCUM maintains consistent performance across different memory constraints. CONTACCUM exhibits robust performance regardless of the memory constraint level (11GB or 24GB), with only minor variations between the two settings. In contrast, the performance of both DPR and GradAccum improves as the available memory increases from 11GB to 24GB. This suggests

[3]https://github.com/Hannibal046/nanoDPR

[4]bert-base-uncased

that the performance gains of CONTACCUM are not significantly affected by the severity of memory limitations.

The effectiveness of CONTACCUM is amplified under more severe memory constraints. While CONTACCUM consistently outperforms the baseline methods in both 11GB and 24GB scenarios, the performance gap between CONTACCUM and the baselines is more substantial in the 11GB setting. This indicates that the advantages of CONTACCUM are particularly evident when memory constraints are stringent, emphasizing its effectiveness in low-resource setting. The strong performance of CONTACCUM can be attributed to its dual memory bank strategy, which allows it to utilize more negative samples than GradCache, even in low-resource settings. Furthermore, CONTACCUM outperforms the high-resource setting in 18 out of 24 metrics, improving up to 4.9 points. In contrast, GradCache only surpasses the high-resource setting in 8 metrics, with marginal improvements likely due to randomness. These results demonstrate the fundamental advantage of CONTACCUM in achieving superior performance compared to both the baselines and the high-resource setting.

5.2 Influence of each components in CONTACCUM

Table 2: Results of removing the components of CONTACCUM. The DPR performance in low-resource (BSZ=8) and high-resource (BSZ=128) settings are shown as baselines. The best-performing method is highlighted in bold.

w/ Hard Negative		w/o Hard Negative	
Method	**Top@20**	**Method**	**Top@20**
DPR (BSZ=8)	70.9	DPR (BSZ=8)	63.7
DPR (BSZ=128)	78.4	DPR (BSZ=128)	74.3
CONTACCUM (ours)	**78.8**	**CONTACCUM (ours)**	**76.3**
w/o. M_q	70.8	w/o. M_q	72.3
w/o. Past Enc.	76.5	w/o. Past Enc.	73.4
w/o. M_q/Past Enc.	67.8	w/o. M_q/Past Enc.	73.9
w/o. GradAccum	76.7	w/o. GradAccum	74.1

Table 2 shows the influence of key components in CONTACCUM by removing each component with NQ. We also reported experiments that excluded hard negatives during training to observe the tendency. The most significant performance drop occurred when the query memory bank M_q was removed, indicating its crucial role in CONTACCUM. The other components of CONTACCUM also contributed to the overall performance, with consistent trends regardless of using hard negatives.

Passage memory bank alone degrades performance due to gradient norm imbalance. Specifically, using only the passage memory bank (w/o. M_q), similar to the pre-batch negatives, led to an 8-point performance drop in Top@20 compared to CONTACCUM. This decrease can be attributed to the gradient norm imbalance problem highlighted in Section 3.3. Section 5.5 further analyzes this issue.

GradAccum and past encoder representations are crucial for stable training and performance. Moreover, when GradAccum was not applied (w/o. GradAccum), a 2.1-point performance decline was observed in Top@20, highlighting the importance of involving more data in gradient calculations for stable training in CONTACCUM. Additionally, a 2.3-point performance decrease was noted when representations generated by past encoders were not used (w/o. Past Enc.). This finding confirms that past encoder representations contribute to training, as suggested by previous studies [37, 38, 19]. However, unlike pre-batch negatives, query memory bank M_q demonstrates that the greatest performance improvement is achieved by employing a dual memory bank, which leverages representations generated by past query and passage encoders.

5.3 Memory bank size analysis

Figure 3 indicates the experimental results on the NQ dataset, demonstrating the impact of memory bank size N_{memory} and accumulation steps K on CONTACCUM's performance in a low-resource setting with a local batch size of 8. As the memory bank size N_{memory}increases, more negative passages are utilized in training, and as the accumulation steps increase, more data is considered in each model update. The performance of DPR in both low-resource and high-resource scenarios($N_{\text{total}} =$

 https://doi.org/10.52202/079017-0375

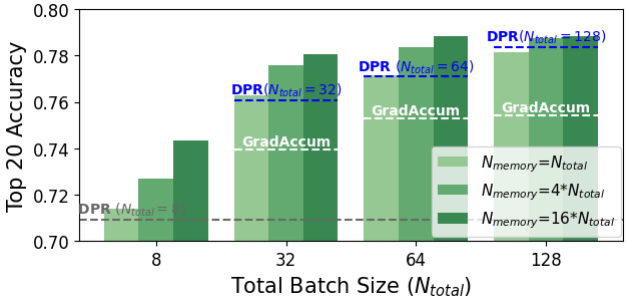


Figure 3: Analysis of accumulation step and memory bank size. DPR performance in low-resource (BSZ=8) and high-resource (BSZ=128) settings is shown as baselines, along with the performance of gradient accumulation for each total batch size (N_{total}).

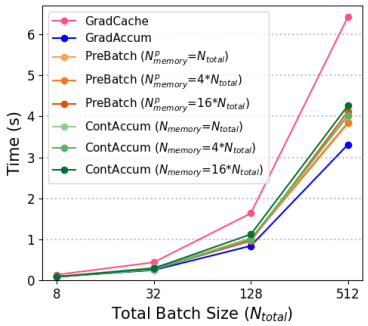


Figure 4: Comparison of the speed of one weight update for different methods as the total batch size (N_{total}) changes.

$32, 64, 128$) is also included for comparison. Note that gradient accumulation is not used when the total batch size is 8 and only the dual memory bank is employed.

CONTACCUM consistently outperforms GradAccum and DPR regardless of the size of memory bank and accumulation step. The results show that increasing the memory bank size improves performance even when GradAccum is not used. This indicates that even without gradient accumulation, utilizing representations from the memory bank to construct a larger similarity matrix enhances performance. This trend remains consistent as the accumulation step increases. Moreover, CONTACCUM consistently outperforms GradAccum in all N_{total} settings. Remarkably, CONTACCUM with $N_{\text{local}} = 8$, $N_{\text{total}} = 64$, and $N_{\text{memory}} = 128$ surpasses the performance of DPR in a high-resource setting ($N_{\text{total}} = N_{\text{local}} = 128$). The performance improvement of CONTACCUM converges as the accumulation step and memory bank size increase, demonstrating that CONTACCUM can robustly enhance performance regardless of memory bank size and accumulation steps.

5.4 Train speed

In this subsection, we compare the training speed of CONTACCUM with baseline methods. Figure 4 shows the results of experiments comparing the speed of a single training iteration (1 weight update) as the accumulation step increases in a low-resource settings with 11GB of available memory. Unlike the high-resource setting, where the total batch can be processed through forward and backward pass at once, the train speed slow down in low-resource settings due to various computations and storing gradients.

CONTACCUM achieves faster iteration times than GradCache, even with large memory banks. As shown in Figure 4, CONTACCUM performs single iterations faster than GradCache in all total batch size. Notably, when $N_{\text{total}} = 512$, GradCache is 93% slower than GradAccum, while CONTACCUM only takes 26% more time, even with the largest memory bank size of $N_{\text{memory}} = 8192$. This indicates that CONTACCUM completes iterations 34% faster than GradCache. The significant additional time for computing one iteration in GradCache is due to the overhead of calculating and storing gradients of representations, as well as the repetitive forward and backpropagation. In contrast, CONTACCUM incurs a relatively minor loss of speed compared to GradAccum due to the additional computations involved in storing and retrieving representations from the memory bank and calculating the enlarged similarity matrix. While pre-batch negatives [19] shows similar computational efficiency to our method, it degrades the performance as demonstrated in Table 2.

5.5 Gradient norm ratio

We conducted experiments comparing the gradient norms of the query and passage encoders to investigate whether the presence of a query memory bank M_q affects the *gradient norm imbalance problem*, as discussed in Section 3.3. The results are presented in Figure 5. This experiment defines the ratio of gradient norms between the two encoders as *GradNormRatio* $= ||\nabla_\Lambda||_2 / ||\nabla\Theta||_2$. We

https://doi.org/10.52202/079017-0375

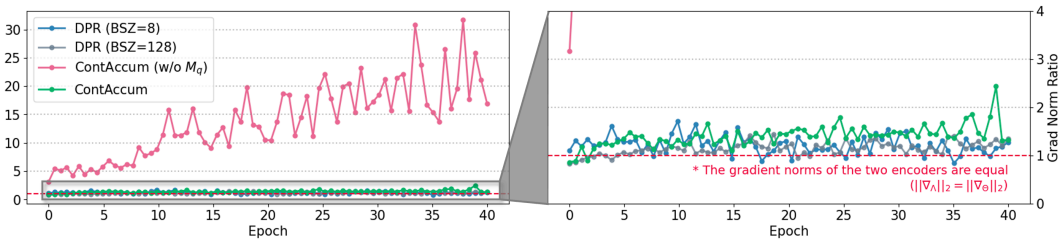


Figure 5: Analysis of GradNormRatio throughout the training process on the NQ dataset.

measured GradNormRatio during the training of the NQ.[5] If the two encoders have similar gradient norms during training, GradNormRatio should be close to 1. If the passage encoder (g_Λ) has a larger gradient norm, GradNormRatio will be greater than 1.

Dual memory bank helps maintain gradient norm balance. The experimental results show that when the query memory bank M_q is not used, GradNormRatio consistently increases. In contrast, CONTACCUM, which utilizes a dual memory bank (M_q, M_p), maintains a GradNormRatio close to 1, similar to DPR.

This indicates that the pre-batch negatives exhibit *gradient norm imbalance problem*. It is because pre-batch negatives only use passage memory bank, leading to an imbalance in the number of query and passage representations used in gradient calculations, as discussed in 3.3. The *gradient norm imbalance problem* consistently occurred even when the timing of omitting the query memory bank M_q is varied during training, as shown in Figure 6.

The *gradient norm imbalance problem* observed during the actual training process becomes increasingly severe, causing the gradient norm of the passage encoder to be up to 30 times larger than the query encoder. As noted by Senushkin et al. [33] and Chen et al. [4], such extreme differences in gradient norms between the two models negatively impact performance. The significant performance drop observed in 5.2 when the query memory bank M_q is not used can be attributed to the *gradient norm imbalance problem*.

6 Conclusion

In this work, we proposed CONTACCUM, a novel memory reduction methodology for training dual-encoders with InfoNCE Loss in low-resource settings. By employing a dual memory bank structure, CONTACCUM achieves stable training and outperforms high-resource baselines, as demonstrated through extensive experiments on five information retrieval datasets. Our mathematical analysis of the dual-encoder training process underscores the importance of balanced gradient norms, which is effectively addressed by the dual memory bank approach. Furthermore, various ablation experiments showed that the accumulation step and memory bank size significantly contribute to performance improvement.

Limitations. While CONTACCUM reduces computational costs and stabilizes training, this study is limited by its focus on supervised fine-tuning. Recently, many studies have proposed a pre-training stage for dense retriever [12, 7, 8, 29]. It remains to be investigated whether the *gradient norm imbalance problem* arises during the pre-training stage and whether CONTACCUM can alleviate it. Additionally, CONTACCUM still relies on the softmax operation, which incurs high computational costs. Reducing this reliance on the softmax operation could lead to more efficient training and broader application of the dense retriever.

Broader impacts. CONTACCUM is designed to train dense retrievers efficiently, which allows it to be applied to various knowledge-intensive systems with limited resources. Examples of such applications include search engines, retrieval-augmented generation, and fact verification on local machines. However, we strongly discourage the use of CONTACCUM in high-risk domains such as medical and legal fields, where the retrieval of incorrect information could have a serious impact.

[5]The values of gradient norms are recorded after gradient clipping.

https://doi.org/10.52202/079017-0375

Future works. In future work, we plan to extend CONTACCUM to the pre-training phase with a uni-encoder structure to assess its broader applicability. We also aim to investigate efficient training strategies to mitigate the substantial computational burden caused by the softmax operation. By addressing these areas, we hope to encourage further research on optimizing dual-encoder training for low-resource settings in the field of information retrieval.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00407803). This work was also supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00460011, Climate and Environmental Data Platform for Enhancing Climate Technology Capabilities in the Anthropocene (CEDP).
We would like to express our sincere gratitude to Keonwoo Kim, Joonwon Jang, Hyowon Cho, Minjin Jeon, and Sangyeop Kim for their valuable feedback and insightful comments. We also deeply appreciate our collegues; Joonghoon Kim, Saeran Park, SangMin Lee, Jiyoon Lee, Jaewon Cheon, and Seonghee Hong - for their constructive discussions and support throughout this work.

References

[1] Petr Baudiš and Jan Šedivý. 2015. Modeling of the Question Answering Task in the YodaQA System. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228, Cham. Springer International Publishing.

[2] Jonathan Berant, Andrew K. Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Conference on Empirical Methods in Natural Language Processing*.

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*.

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 794–803. PMLR.

[5] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *The Eleventh International Conference on Learning Representations*.

[6] 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.

[7] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[8] Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

[9] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP*.

[10] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

[11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, Los Alamitos, CA, USA. IEEE Computer Society.

[12] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.

[13] Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, Suhang Wang, Jiawei Han, and Xianfeng Tang. 2023. Language Models As Semantic Indexers.

[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

 https://doi.org/10.52202/079017-0375

[15] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

[17] Brian Kulis. 2013. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364.

[18] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

[19] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning Dense Representations of Phrases at Scale. In *Association for Computational Linguistics (ACL)*.

[20] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

[21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning*.

[22] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *Advances in Neural Information Processing Systems*, volume 34, pages 9694–9705. Curran Associates, Inc.

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[24] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2791, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[25] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2023. Nonparametric Masked Language Modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada. Association for Computational Linguistics.

[26] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[28] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

[29] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to Retrieve Passages without Supervision. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2687–2700, Seattle, United States. Association for Computational Linguistics.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[31] Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. PAIR: Leveraging Passage-Centric Similarity Relation for Improving Dense Passage Retrieval. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2173–2183. Association for Computational Linguistics.

[32] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

[33] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. 2023. Independent Component Alignment for Multi-Task Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20083–20093.

[34] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2023. Learning to Tokenize for Generative Retrieval. In *Advances in Neural Information Processing Systems*, volume 36, pages 46345–46361. Curran Associates, Inc.

[35] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

[36] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748.

[37] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-Batch Negative Sampling for Training Two-Tower Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1632–1636, New York, NY, USA. Association for Computing Machinery.

[38] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. 2020. Cross-Batch Memory for Embedding Learning. In *CVPR*.

[39] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. 2018. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, Los Alamitos, CA, USA. IEEE Computer Society.

[40] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. 2018. Improving Generalization via Scalable Neighborhood Component Analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[41] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

 https://doi.org/10.52202/079017-0375

[42] Zhen Yang, Zhou Shao, Yuxiao Dong, and Jie Tang. 2024. TriSampler: A Better Negative Sampling Principle for Dense Retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):9269–9277.

[43] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense Text Retrieval Based on Pretrained Language Models: A Survey. *ACM Trans. Inf. Syst.*, 42(4).

A. Derivatives of InfoLoss with memory bank

$$\nabla_\Theta \mathcal{L}(S_k) = \sum_{\mathbf{q}_l \in Q_k^t} = \frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{q}_l} \cdot \frac{\partial \mathbf{q}_l}{\partial \Theta}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{q}_l} &= \frac{\partial}{\partial \mathbf{q}_l}(-\frac{1}{N_{\text{local}} + N^q_{\text{memory}}}) \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \log \frac{\exp(\mathbf{q}_i \cdot \mathbf{p}_i^\top)}{\sum_j^{N_{\text{local}}+N^I_{\text{memory}}p} \exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top)} \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \left[\frac{\partial}{\partial \mathbf{q}_l}(\mathbf{q}_i \cdot \mathbf{p}_i^\top) - \frac{\partial}{\partial \mathbf{q}_l} \log \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top) \right] \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{p}_l - \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \left[\frac{\exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top)}{\sum_k^{N_{\text{local}}+N^p_{\text{memory}}} \exp(\mathbf{q}_i \cdot \mathbf{p}_k^\top)} \cdot \frac{\partial}{\partial \mathbf{q}_j} (\mathbf{q}_i \cdot \mathbf{p}_j^\top) \right] \right) \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{p}_l - \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \left[S_{k(i,j)} \cdot \frac{\partial}{\partial \mathbf{q}_l}(\mathbf{q}_i \cdot \mathbf{p}_j^\top) \right] \right) \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{p}_l - \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \left[S_{k(l,j)} \cdot \mathbf{p}_j \right] \right)
\end{aligned}$$

$$\nabla_\Lambda \mathcal{L}(S_k) = \sum_{\mathbf{p}_l in P_k^t} = \frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{p}_l} \cdot \frac{\partial \mathbf{p}_l}{\partial \Lambda}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(S_k)}{\partial \mathbf{p}_l} &= \frac{\partial}{\partial \mathbf{p}_l}(-\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \log \frac{\exp(\mathbf{q}_i \cdot \mathbf{p}_i^\top)}{\sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top)} \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \left[\frac{\partial}{\partial \mathbf{p}_l}(\mathbf{q}_i \cdot \mathbf{p}_i^\top) - \frac{\partial}{\partial \mathbf{p}_l} \log \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top) \right] \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{q}_l - \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \left[\frac{\exp(\mathbf{q}_i \cdot \mathbf{p}_j^\top)}{\sum_k^{N_{\text{local}}+N^p_{\text{memory}}} \exp(\mathbf{q}_i \cdot \mathbf{p}_k^\top)} \cdot \frac{\partial}{\partial \mathbf{p}_l (\mathbf{q}_i \cdot \mathbf{p}_j^\top)} \right] \right) \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{q}_l - \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} \sum_j^{N_{\text{local}}+N^p_{\text{memory}}} \left[S_{k(i,j)} \cdot \frac{\partial}{\partial \mathbf{p}_l}(\mathbf{q}_i \cdot \mathbf{p}_j^\top) \right] \right) \\
&= -\frac{1}{N_{\text{local}} + N^q_{\text{memory}}} \left(\mathbf{q}_l - \sum_i^{N_{\text{local}}+N^q_{\text{memory}}} S_{k(i,l)} \cdot \mathbf{q}_l \right)
\end{aligned}$$

B. Details on hyperparameters

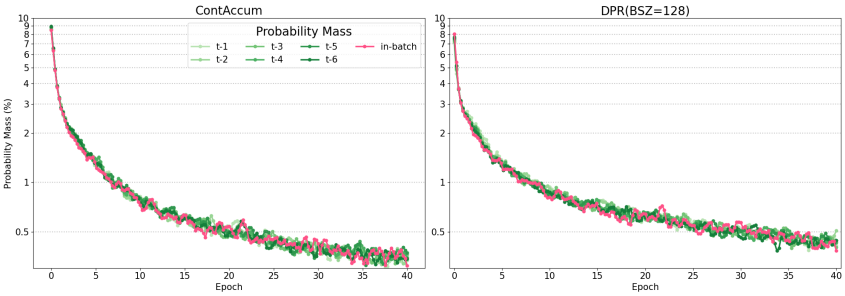
Hyperparameters. The hyperparameters for training were set as follows: the warmup step was 1,237 steps, weight decay was set to 0, and a customized scheduler with a linear decay of the learning rate after the warmup was used. The optimizer was AdamW [23] with epsilon set to 1e-8, and the learning rate was 2e-5. Gradient clipping was applied at a value of 2.0, and τ was set to 1. For retrieval, we used the FAISS [14] library to perform exact nearest neighbor search with default hyperparameters.

C. Similarity Mass

To verify whether representations generated by past encoders aid the current encoder's training, we conducted an experiment measuring the similarity mass of passage representations at different time

https://doi.org/10.52202/079017-0375

Figure 6: Experiments on similarity probability mass.



steps. The results are shown in Figure 6. The similarity mass is defined as the sum of similarities after passing through a softmax function for all current time t queries with passage representations generated at past time steps $t-k$, as shown in Equation 10:

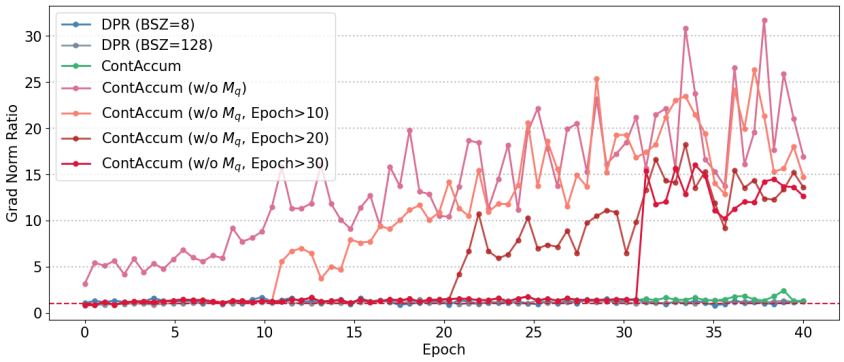
$$\text{SimMass}_{t-k} = \frac{1}{|\mathbf{Q}^t|} \sum_{i=1}^{|\mathbf{Q}^t|} \sum_{j=1}^{|\mathbf{P}^{t-k}|} \mathbf{Q}_i^t \cdot (\mathbf{P}_j^{t-k})^\top \tag{10}$$

Passage representations of the current and previous encoder have similar importance as negative passage. The results indicate that there is no significant difference in the similarity mass between the in-batch negative passage representations at the current training step and the passage representations from up to six previous steps. As shown in Equation 8 and 9, the gradients of the two encoders are proportional to the magnitude of the similarities. This means that negative passages with high similarity to a single query produce large gradients, which aids in training the dense retrieval model [41]. This finding suggests that past representations can be beneficial from the early stages of training, contrary to previous studies [37, 38].

Additionally, as illustrated in Figure 6, CONTACCUM demonstrates the same similarity mass trend as DPR, validating the effectiveness of utilizing past representations from the early stages of training with CONTACCUM.

D. Gradient Norm Ratio of Omitting the Query Memory Bank

Figure 7: Experimental results of omitting query memory bank during training.



***Gradient norm imbalance problem* occurs when the query memory bank is omitted**. We omitted the query memory bank during training at various epochs: [10, 20, 30]. As shown in Figure 7, the gradient norm imbalance problem arises immediately after the query memory bank is excluded. Additionally, irrespective of when the query memory bank is omitted, all experiments without the query memory bank exhibit very high gradient norm ratios in the later stages of training. This indicates that *gradient norm imbalance problem* can cause unstable training during the entire training process, unlike previous studies which mentioned the major cause of unstable training is rapid changes in encoder representations in the early epochs [38, 37].

https://doi.org/10.52202/079017-0375

E. Actual Memory Usage

ContAccum uses few memory for dual memory bank but it works greatly Theoretically, CONTACCUM's query and passage memory banks(M_q, M_p) do not cache activation values, requiring only additional memory for the stored representations compared to GradAccum. The memory usage of the dual memory bank can be calculated as follows:

$$\mathbf{N}_{\text{memory}} \times \text{dim}_{\text{embed}} \times 2 \times 4 \tag{11}$$

where 2 represents the query and passage memory bank and 4 denotes full precision (4 bytes).

Table 3: Comparison of Memory Usage

Method	$N_{\textbf{local}}/K/N_{\textbf{total}}/N_{\textbf{memory}}$	**Memory (GB)**	**Additional Memory**	
			Actual	Theoretical
DPR	8/ 1/128/ 0	7.483	-	-
GradCache	8/16/128/ 0	5.158	-	-
GradAccum	8/16/128/ 0	8.340	-	-
CONTACCUM	8/16/128/ 128	8.342	0.002	0.0007
CONTACCUM	8/16/128/ 512	8.346	0.006	0.0029
CONTACCUM	8/16/128/1024	8.353	0.013	0.0059
CONTACCUM	8/16/128/5096	8.382	0.042	0.0117

Moreover, we measured each method's actual memory usage in a VRAM=11GB environment and the results are reported in table 3. The results show that CONTACCUM uses only up to 0.5% more memory than GradAccum, which is a maximum of 12MB even in the largest memory bank size($\mathbf{N}_{\text{memory}} = 5096$). This demonstrates that CONTACCUM is a memory-efficient method that consumes very limited additional memory compared to GradAccum. Furthermore, while GradCache uses less memory than DPR by decomposing complex forward and backward processes, it has the limitation of very slow training speed, as shown in Figure 4.

F. License

The licenses for the assets used in this paper are as follows:

- Overall train code and partial evaluation code from nano-DPR: CC-BY-NC 4.0
- Train and evaluation datasets preprocessed by DPR: CC-BY-NC 4.0
- Partial evaluation code, and train and evaluation dataset preprocessed by Beir: Apache-2.0
- Hard negative score generated by sentence-transformers library: Apache-2.0

https://doi.org/10.52202/079017-0375

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims of this paper accurately represent its contributions and scope. The abstract and introduction clearly outline the proposed methods and expected outcomes, which are consistently supported by the results and discussions presented in the subsequent sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discusses the limitations of the work in the Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This paper provide the full set of assumptions and a complete proof in Section 3.1 for each theoretical result provided in Section 5.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper fully discloses all the information needed to reproduce the main experimental results. Specifically, the datasets and evaluation metrics are provided in Section 4, and the details of the proposed algorithm (CONTACCUM) are given in Section 3.1. Additionally, information about the hyperparameters for the training dynamics (e.g., optimizer, learning rate, scheduler, etc.) is provided in Appendix 6.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

https://doi.org/10.52202/079017-0375

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All codes and links to download the datasets are included in the supplemental material. Additionally, we plan to release the codes for reproducibility of the main experimental results after the review process to preserve anonymity.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specify all the training and test details in the Section 4 and Appendix 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the high computation cost of the experiments, this paper does not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The entire set of experiments was conducted using the same computer resources described in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Potential Positive and negative societal impacts are discussed in Section 6.

https://doi.org/10.52202/079017-0375

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper only proposes an efficient training algorithm for information retrieval, so it poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All creators or original owners of assets are clarified in Section 4. Also, the license of each assets are mentioned in Appendix 6.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/079017-0375

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.