Query-Only Attention for Trustworthy Continual Adaptation

Gautham Udayakumar Bekal
Independent Researcher
gautham.bekal@enlyte.com

Ashish Pujari
University of North Carolina at Charlotte
apujari1@uncc.edu

Scott David Kelly
University of North Carolina at Charlotte
skelly52@charlotte.edu

Abstract

Foundation models deployed in dynamic environments face continual distributional shifts and evolving data conditions, where failure to adapt can erode reliability and fairness. We propose a Query-Only Attention mechanism that discards keys and values while preserving the inductive bias of full-attention architectures. In continual learning scenarios, this simplified mechanism significantly mitigates both loss of plasticity and catastrophic forgetting, outperforming baselines such as selective re-initialization. Query-Only Attention achieves competitive performance to full attention while being more compute-efficient. We establish a conceptual link between query-only attention, full transformer attention, and model agnostic meta-learning, framing them as instances of meta-learning. Finally, through Hessian-spectrum analysis, we show that models maintaining higher curvature rank across tasks exhibit sustained adaptability, improving trustworthiness under distribution shift. These findings highlight principles relevant to real-world continual learning systems that demand reliability, fairness, and accountability.

1 Introduction

Continual learning remains a fundamental challenge in deep learning [25], where models must learn from non-stationary data streams without succumbing to catastrophic forgetting [9] or loss of plasticity [3]. For foundation models deployed in dynamic environments, failure to adapt under distributional shift can silently degrade reliability, fairness, and safety, making continual adaptation central to trustworthy and responsible AI.

While existing approaches mitigate forgetting using replay buffers [28] or regularization [9], preserving plasticity—the ability to acquire new knowledge—remains less explored. Recent studies show that transformer attention [22] inherently supports plasticity across tasks [24]. Motivated by this, we ask: Can the core mechanism of attention be simplified for scalable and trustworthy continual adaptation?

We propose Query-Only Attention (QOA), a minimalist mechanism that removes keys and values yet preserves the inductive bias of full attention. QOA maintains stable curvature (high Hessian rank) across tasks, enabling controlled learning and unlearning—a property vital for safe knowledge editing and continual fine-tuning of foundation models. Our analysis links QOA, full attention, and Model-Agnostic Meta-Learning [6], showing that both achieve stable, near-constant curvature indicative of robust adaptability.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Socially Responsible and Trustworthy Foundation Models (NeurIPS 2025).

In safety-critical domains such as healthcare or public policy, responsible foundation models must adapt continually without amplifying bias, compromising privacy, or degrading performance. QOA contributes toward this goal by combining efficient continual adaptation with trustworthy behavior.

Key contributions:

- We introduce *Query-Only Attention*, which is based on attention mechanism and meta-learning that mitigates loss of plasticity more effectively than state-of-the-art methods in fully online continual learning experiments.
- As a natural consequence, Query-Only Attention also mitigates catastrophic forgetting when task identity is available, although *forgetting reduction is not the primary focus of this work*.
- We provide a conceptual explanation showing that Query-Only Attention mitigates both loss of plasticity and forgetting by converging toward a *global* rather than task-specific *local* solution.
- We establish conceptual links between Query-Only Attention, the original attention mechanism [22], and meta-learning approaches such as MAML [6] in continual learning context.
- We analyze the Hessian spectrum and effective rank [13], demonstrating that decreasing rank across tasks correlates with loss of plasticity.

2 Related work

Deep neural networks have shown remarkable generalization capabilities on unseen tasks. However, they typically operate under the assumption that the training data is stationary and that all samples are available simultaneously during training. In contrast, online learning assumes that data arrives sequentially in a stream and each data point is observed only once, eliminating the concept of epochs. In such a setting, the model must continuously update its parameters to adapt to incoming data. From the model's perspective, the data distribution is inherently non-stationary, since not all samples are available at the same time. This leads to two major challenges: catastrophic forgetting and loss of plasticity. Catastrophic forgetting — the degradation of performance on previously learned tasks after training on new ones — has been extensively studied in the literature [15], [8], [18]. A more fundamental and less studied issue is loss of plasticity, the gradual reduction in a model's ability to learn new tasks altogether [3], [16]

Hence, continual learning faces challenge on two fronts, forward performance / mitigating loss of plasticity and backward performance / mitigating catastrophic forgetting. [2] showed that there exists a tradeoff between the two. We show an alternative view that Query-Only Attention and related architectures attain a global solution which mitigates both loss of plasticity and catastrophic forgetting simultaneously.

Most papers, analyze one of the two of above challenges, however very few papers tackle both challenges simultaneously. Regularization-based continual learning methods such as Elastic Weight Consolidation (EWC) [9], [27], and Learning without Forgetting [14] were designed primarily to mitigate catastrophic forgetting, i.e., the degradation of previously learned tasks. However, these approaches do not directly address the complementary challenge of loss of plasticity, where the model fails to acquire new tasks altogether. Our focus in this work is specifically on loss of plasticity, where forgetting is a downstream consequence rather than the central phenomenon. For this reason, we compare against baselines that are explicitly targeted at plasticity, including [3] and recent attention-based approaches [24]. For completeness, we also evaluate an EWC-style method on one benchmark.

One such paper is [5] which uses utility based methodology for handling both catastrophic forgetting and loss of plasticity. Here, the authors are working with unknown task boundaries. However, obtaining the utility is expensive especially in the era of large and very large models. Controlling gradient updates based on weight utility leads to reduced ability to retain old tasks as their number increases. Most importantly, this method is ad-hoc solution for continual learning problem and not a more global solution which can scale to very large number of tasks. Our method contrasts in that it obtains a global solution and thus has no reduction in performance irrespective number of tasks.

The core algorithm we developed is most closely related to the paper [24] which utilizes attention network and replay buffer to handle this challenge. However, attention network has $O(n^2)$ in compute

which can be challenging in continual learning setting where data will come rapidly. Here, n is the size of replay buffer. We draw our inspiration from this paper on using replay buffer but make a novel hypothesis that query matrix is all that is needed for continual learning. Our method achieves similar or superior performance compare to full attention and can also do the compute in O(n). The other aspect being [24] does not explain the intriguing phenomenon. Here we carry out a detailed empirical and theoretical analysis on why query-only attention works.

Our work reveals deep connection between attention network, in-context learning [4], [26], [1] model agnostic meta learning [6], and metric based meta learning [23], [10], [20] under the paradigm of continual learning.

To analyze plasticity in continual adaptation, we measure the effective Hessian rank [13, 19], where higher rank indicates sustained adaptability. Query-Only Attention maintains a stable or non-decreasing rank across tasks, supporting reliable, transparent, and controllable learning dynamics essential for trustworthy foundation models.

3 Background and preliminaries

Our theoretical analysis builds on several standard components: attention networks, meta-learning (in particular MAML), and k-nearest neighbors (KNN). We briefly review each here to fix notation and highlight the connections that will be used in Section 6.

3.1 Attention mechanism

In the standard attention network [22], each query vector q_i produces a weighted combination of value vectors $V = \{v_1, \ldots, v_n\}$:

$$\text{Attn}(q_i, K, V) \;=\; \sum_{j=1}^{n} \alpha_{ij}\, v_j, \tag{1}$$

where the weights α_{ij} are obtained from a softmax over query–key dot products:

$$\alpha_{ij} \;=\; \frac{\exp\left(q_i^\top k_j/\sqrt{d}\right)}{\sum_{j'=1}^{n} \exp\left(q_i^\top k_{j'}/\sqrt{d}\right)}. \tag{2}$$

Here $K = \{k_1, \ldots, k_n\}$ are key vectors and d is the feature dimension. This requires computing all pairwise dot products $q_i^\top k_j$, which scales as $\mathcal{O}(n^2)$ in sequence length n. In contrast, our query-only model removes K and V, learning task-specific weights θ' directly, while still preserving the interpretation of predictions as weighted combinations over context points.

3.2 Meta-learning

Meta-learning aims to enable rapid adaptation to new tasks from a few support examples [6]. Traditional meta-learning assumes task boundaries and episodic training, and is thus non-continual. Recent work [7] , [21] explores meta-learning for continual learning. Our approach draws inspiration from meta-learning while operating fully online.

Model-agnostic meta-learning
Following Finn et al. [6], MAML optimizes model parameters via alternating *inner-loop* and *outer-loop* updates. The inner loop computes task-specific parameters Θ_i' using the support set, while the outer loop updates the shared parameters Θ based on query-set performance:

$$\Theta_i' = \Theta - \alpha \nabla_\Theta \mathcal{L}_\Theta(\text{support}_i), \tag{3}$$

$$\Theta \leftarrow \Theta - \beta \sum_{i=1}^{m} \nabla_\Theta \mathcal{L}_{\Theta_i'}(\text{query}_i). \tag{4}$$

Here, Eq. 3 defines task-specific adaptation, and Eq. 4 updates the meta-parameters shared across tasks.

3.3 k-Nearest neighbors (KNN)

In k-nearest neighbors regression, prediction is based on the k closest points in a support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ under a distance metric $d(\cdot, \cdot)$ (e.g., Euclidean).

Given a query input x, let $\mathcal{N}_k(x)$ denote the indices of the k nearest neighbors. The kNN prediction is the local average:

$$\hat{y}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i. \tag{5}$$

Thus, kNN regression is a non-parametric, memory-based method where predictions adapt directly from nearby support examples.

3.4 In-context learning

In-context learning has recently been interpreted as an implicit k-nearest-neighbors mechanism that emerges in the forward pass of transformers [17]. This perspective provides a key motivation for our work: by modifying attention, we aim to enable continual adaptation without requiring explicit task identifiers.

4 Problem statement

We study the *continual learning (CL)* setting, where a model receives a continuous stream of data. Each data point is observed *once* during training, without repetition or epochs.

Formally, the stream is generated from a sequence of tasks $\{T_1, T_2, \ldots, T_n\}$. Each task T_i is associated with a (potentially non-stationary) distribution $\mathcal{D}_i(x, y)$ over input–label pairs (x, y).

Training.

- The model does not observe task boundaries or task identities.
- Samples arrive sequentially, drawn from the evolving distribution.
- The objective is to update the model online while retaining performance across all tasks.
- Based on the task at hand, the model may update on a single data point or a batch of data-points.

Evaluation Protocol. During inference, the model processes a data stream sequentially. At data point m of task t, the goal is to predict the next n points $\{m+1, \ldots, m+n\}$ from the same task. The resulting accuracy (or loss) defines the *forward performance*; its degradation over time indicates *loss of plasticity*.

After training up to task t, the model is also evaluated on samples from a previous task $t - j$. The resulting accuracy defines the *backward performance*, and its degradation quantifies *catastrophic forgetting*.

- **Forward testing (plasticity):** Evaluates on upcoming data from the current stream without task identifiers. A decline in this metric across tasks signals loss of plasticity.
- **Backward testing (forgetting):** Evaluates on a small held-out buffer of past-task samples where task identities are known. A decline in this metric indicates catastrophic forgetting.

5 Method

5.1 Query only model with replay buffer

Drawing the connection from attention networks, meta-learning, KNN and replay buffer we design an architecture which obtains an optimal solution for continual learning task, leading to mitigation of both

loss of plasticity and catastrophic forgetting simultaneously. A sample data point is $d = (x, y) \in \mathcal{D}$. Let, $x \in \mathbb{R}^a$ and $y \in \mathbb{R}^b$ We define a buffer $\mathcal{B}$ containing n data-points. Every time step we construct a support set $\mathcal{S}$ of size m sampled from $\mathcal{B}$ such that $m \leq n$ depending on the problem at hand. Hence, $S \in \mathbb{R}^{m*(a+b)}$

$$S = \begin{bmatrix} x_{s1} & y_{s1} \\ x_{s2} & y_{s2} \\ \vdots & \vdots \\ x_{sm} & y_{sm} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}$$

Let, the data-point on which we want to make prediction be query q_i, such that $q_i = x_i$. The neural-net model is a single query-matrix $\mathcal{Q}_\theta \in \mathbb{R}^{(2a+b)*b}$, for illustration purposes and can have multiple layers.

Algorithm 1 Query-Only Attention with Replay Buffer

Input: Stream of tasks $\{T_1, T_2, \ldots\}$; replay buffer $\mathcal{B}$ of size n; support size m; query-only model Q_θ; learning rate η
Output: Updated parameters θ
Initialize θ, $\mathcal{B} \leftarrow \emptyset$
for each incoming sample (x_t, y_t) **do**
    Insert (x_t, y_t) into buffer $\mathcal{B}$; evict oldest if $|\mathcal{B}| > n$
    Sample support set $\mathcal{S} = \{(x_j, y_j)\}_{j=1}^m \subset \mathcal{B}$
    Compute scores $d_j \leftarrow Q_\theta(x_t, x_j, y_j)$
    Prediction $\hat{y}_t \leftarrow \sum_{j=1}^m d_j \, y_j$
    Loss $\mathcal{L}_t \leftarrow \ell(\hat{y}_t, y_t)$
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_t$
end for

We can thus write the predictive model as,

$$\hat{y_t}(x) = \sum_{x_i, y_i \in B} Q_\theta(x_t, x_i, y_i) * y_i \tag{6}$$

Here, x_t is the query point, x_i is a support input, and y_i is support label. $Q_\theta(x_t, x_i, y_i)$ denotes a learned similarity (or distance) function. Unlike standard metrics such as dot products or Euclidean distance—which require vectors to be in the same feature space—Q_θ learns a representation where query–support pairs become directly comparable, allowing flexible weighting even when raw dimensions differ.

5.2 MAML with replay buffer

Note. The adaptation of MAML with replay buffer is still work in progress. We include it here to illustrate a potential direction for combining meta-learning with large buffers in continual learning, but do not claim it as a finalized or fully validated algorithm. That said, our preliminary results are promising for one of the experiments, and suggest this variant may provide a complementary approach to attention-based or query-only models.

We adapt MAML to the continual learning setting by introducing a large replay buffer. To minimize interference across tasks, the buffer is evenly partitioned into t sub-buffers, one per sampled task. Without such partitioning, the sampled support and query examples from different tasks would overlap excessively, leading to degraded task separation and unstable meta-updates. Each task provides a small support set (s) and query set (q), which is sufficient for the MAML inner/outer loops.

An important property of this setup is that the support/query sizes remain much smaller than those required by attention-based or query-only models. This makes MAML particularly well-suited for backward evaluation (catastrophic forgetting), since only a small set of examples per task is needed to adapt. Standard MAML inner-loop adaptation and outer-loop meta-updates are then applied on these partitioned tasks. The details of the algorithm are presented in the appendix section.

Algorithm 2 MAML with Replay Buffer (work in progress)

Input: Stream (x_t, y_t); replay buffer $\mathcal{B}$ of size N; tasks t; support s; query q; learning rates α, β

Initialize θ, $\mathcal{B} \leftarrow \emptyset$
for each incoming (x_t, y_t) **do**
    Insert (x_t, y_t) into $\mathcal{B}$; evict oldest if $|\mathcal{B}| > N$
    Partition $\mathcal{B}$ into t equal sub-buffers
    for each task $i = 1 \dots t$ **do**
        Sample s support and q queries from sub-buffer i
        Inner update on support with step size α
        Outer update on queries with step size β
    end for
end for

6 Theoretical discussion

6.1 Global vs local solution

To understand why our algorithm overcomes loss of plasticity and catastrophic forgetting, we first define weighted k-nearest neighbors

$$\hat{y}(x) = \frac{\sum_{i \in N_k(x)} w(x, x_i)\, y_i}{\sum_{i \in N_k(x)} w(x, x_i)}, \tag{7}$$

where $w(x, x_i)$ is a weight function that decreases as the distance $d(x, x_i)$ increases. Common choices include:

$$w(x, x_i) = \frac{1}{d(x, x_i)}, \tag{8}$$

$$w(x, x_i) = e^{-\alpha\, d(x, x_i)}. \tag{9}$$

In Equation 7, predictions depend only on neighboring data points and not on any learnable parameters. Thus, weighted kNN avoids loss of plasticity (no parameters to get stuck in low-rank regions) and catastrophic forgetting (no parameters to overwrite). Performance is fully determined by the support set and chosen distance metric.

Comparing Equation 7 with Equation 6, the difference lies in how the distance metric is obtained: in kNN it is fixed manually, while in the query-only attention it is learned as θ. Once θ is learned, predictions depend primarily on the support set, so continual adaptation occurs in-context rather than through constant parameter updates. This makes the learning global in nature and independent of any single task which is different from vanilla backpropagation or continual backpropagation algorithm where the model updates its parameters continuously local for each task.

6.2 Model agnostic meta-learning for continual learning

We can rewrite equation 6 as,

$$\hat{y}_t(x) = \sum_{x_i, y_i \in B} \theta'_{t,i}\, y_i, \tag{10}$$

From Equation 10, predictions depend on task-specific parameters θ' generated at inference time. This parallels the task-specific adaptation in MAML's inner/outer-loop updates (Equations 3, 4). However, unlike query-only attention models, MAML was originally designed with task IDs and distributions known, an assumption that breaks in continual learning. In our experiments, we find that using a large replay buffer stabilizes MAML training despite this limitation. Even more interestingly, MAML requires a much smaller support set than query-only attention or full attention networks,

which could make it a promising direction for mitigating catastrophic forgetting in future continual learning work where memory efficiency is crucial.

6.3 Relationship to attention network

From Equation 1, the prediction is a linear combination of value vectors, which are themselves transformed representations of the input. Comparing this with Equation 10, we see a strong similarity: both aggregate information from a support set using learned weights. The main difference is that in attention (Equation 2), task specific weights are derived from query–key dot products, whereas in the query-only model they come from a learned distance metric θ'.

This equivalence suggests that attention networks can also mitigate loss of plasticity, much like the query-only model. However, computing attention weights requires all pairwise query–key dot products, leading to $\mathcal{O}(n^2)$ complexity for a support set of size n. In contrast, our query-only model only compares the current query with the support set, reducing the complexity to $\mathcal{O}(n)$. This efficiency allows us to scale to larger support sets in continual learning, improving performance without incurring the prohibitive cost of full attention.

6.4 Hessian rank analysis

To study plasticity, we analyze the *effective rank* of the Hessian [13],[19], defined as

$$\text{erank}(H) = \exp\Big(-\sum_{i=1}^{n} p_i \log p_i \Big), \quad p_i = \frac{\lambda_i}{\sum_j \lambda_j}, \tag{11}$$

where $\{\lambda_i\}$ are the eigenvalues of the Hessian. A stable effective non-decreasing effective rank indicates models that preserve plasticity across tasks.

6.5 Responsible Continual Adaptation

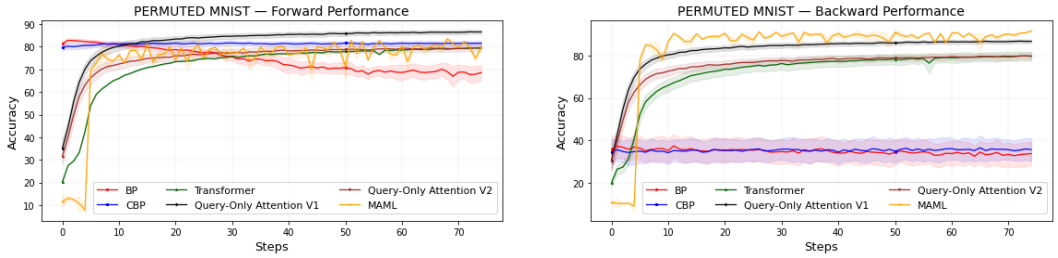
Query-Only Attention maintains a high curvature rank that supports online adaptation and robustness, while also utilizing reduced number of parameters.

7 Experiments

We evaluate forward (plasticity) and backward (forgetting) performance on three benchmarks: **Permuted MNIST** (abrupt shifts), **Tiny ImageNet** [11], and **Slowly Changing Regression (SCR)** (gradual drift). The above experiments follow same setup as explained in Dohare et al. [3], except instead of full ImageNet we choose Tiny ImageNet for efficiency. Distinction is that we perform experiments in a online setting, using *unknown* task boundaries to study loss of plasticity and *known* task boundaries to study catastrophic forgetting. Detailed configurations appear in the Appendix. Results are averaged over three seeds with shaded ± 1 std regions. Higher accuracy (classification) and lower MSE (regression) indicate better performance. Baselines include **BP**, **CBP**, and the **Full-Attention Network**. Since the primary focus of this work is mitigating loss of plasticity, we additionally include the state-of-the-art forgetting baseline **Elastic Weight Consolidation (EWC)** in the ImageNet experiments for completeness. The full details of the experiments is available in Appendix section A.3 , A.4 , A.5

7.1 Permuted MNIST

We evaluate the **query-only attention** model with support sizes of 1000 and 200 and name them as Query-Only Attention V1 and Query-Only Attention V2. The **full-attention** uses support 100, since its $\mathcal{O}(n^2)$ attention limits scalability, while our $\mathcal{O}(n)$ query-only design allows larger supports at lower cost. The **MAML**-style model uses a large replay buffer with a small support of 10, trained on 100 tasks. Each iteration is costlier due to inner-loop updates, so it runs on fewer tasks but performs multiple updates per iteration. For readability, performance curves are averaged over fixed task windows.

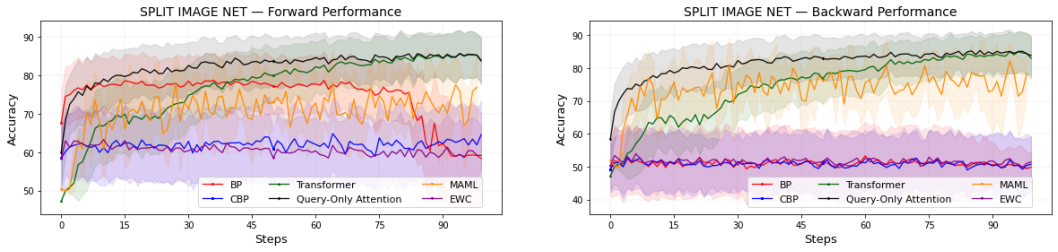


(a) Forward accuracy (permuted MNIST). (b) Backward accuracy (permuted MNIST).

Figure 1: The prediction is over 7500 tasks and each data-point in the graph is averaged over 100 tasks for all models except for MAML. For MAML, we run over only 75 tasks and is shown without averaging.

7.2 Split Image Net

Split-image-net we use a support size of 180 for both query-only attention and full-attention model. For MAML a support size of only size 10 is enough.



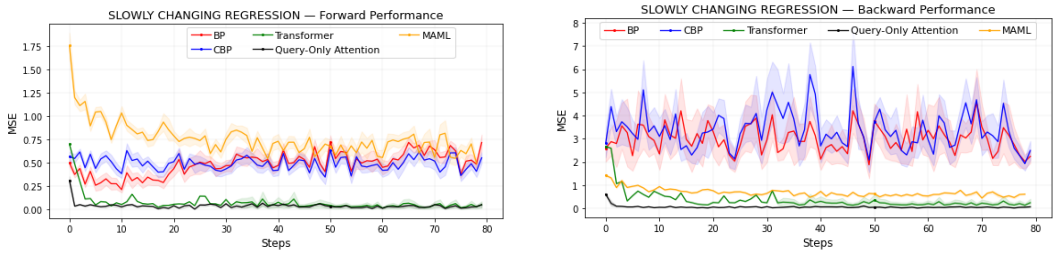
(a) Forward accuracy (Split Image Net). (b) Backward accuracy (Split Image Net).

Figure 2: The prediction is over 9000 tasks and each data-point in the graph is averaged over 100 tasks for all the models except MAML. MAML is run over 500 tasks, averaged over 5 tasks.

Observations in Classification Tasks. Across both 7.1 and 7.2, the **query-only attention** model consistently outperforms all baselines in forward and backward performance. The **full-attention** reaches similar final accuracy but with 50% more parameters and slower convergence. Under unknown task boundaries, **CBP** performs poorly, especially on Split ImageNet, while the **vanilla network** shows clear loss of plasticity. The **MAML**-based model converges fastest, achieving intermediate performance between attention-based and standard networks.

7.3 Slowly Changing Regression

In *SCR*, we use a single query-only attention model with support size 100, *matching* the full-attention (both 100) to isolate algorithmic effects under equal memory/compute. With equal support (100) for query-only attention and full attention, both sustain low MSE in forward testing.



(a) Forward MSE (SCR). (b) Backward MSE (SCR).

Figure 3: The prediction is over 800 tasks and each data-point in the graph is averaged over 10 tasks for all models.

Observations on Regression Task. As in classification, **BP** gradually loses plasticity, while **CBP** fails to learn effectively due to its purely online setup. The **query-only attention** model converges quickly with near-zero MSE, whereas **full attention** converges more slowly with slightly lower

performance. Unlike in classification, the **MAML**-based model struggles to converge but still maintains plasticity. In backward testing, BP and CBP show severe forgetting, while query-only attention, full-attention, and MAML models retain high performance. Including y_i in $Q_\theta(x_t, x_i, y_i)$ offered no gain on this regression task, so we used only (x_t, x_i); for Permuted-MNIST, label inclusion improved results and was retained.

Result analysis. Across all benchmarks, **query-only attention** and **full-attention** models perform similarly, but the query-only attention model converges faster and scales better with $\mathcal{O}(n)$ complexity. The **MAML**-based approach shows strong backward performance and quick convergence with minimal support, though less consistent on SCR. **Query-only attention** model mitigates loss of plasticity and forgetting, highlighting its practicality under limited compute and memory. These findings suggest that preserving plasticity contributes directly to the reliability and interpretability of model updates, ensuring safer adaptation under non-stationary data distributions.

7.4 Effective Rank

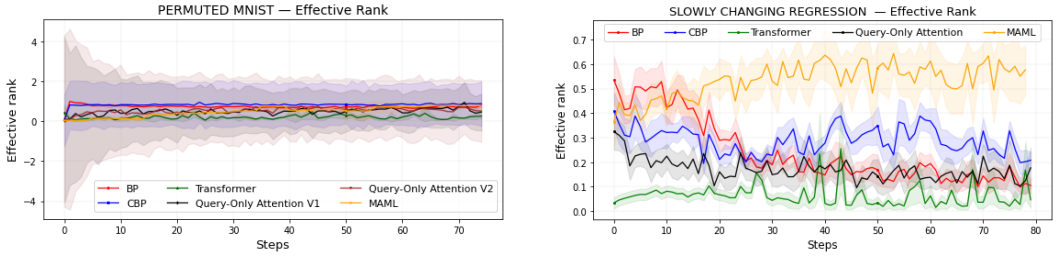


Figure 4: Effective rank co-varies with forward performance; dips align with reduced plasticity. Mean $\pm$ std over 3 seeds.

The effective rank provides a measure of model plasticity. We measure it at the start of each task using the Hessian of the final layer only, since full-Hessian computation is infeasible. We measure it only on Permuted MNIST and SCR and not on image-net due to computational constraints. Also, the effective rank has been normalized since effective rank will depend on size of neural net. In both Permuted-MNIST and SCR, vanilla backpropagation shows a steady drop in effective rank, aligning with loss of plasticity. All other models show near minimal drop in effective rank thus indicating preserved plasticity.

8 Conclusion

We introduced a Query-Only Attention mechanism for continual learning, showing that it mitigates loss of plasticity even when task boundaries are unknown, outperforming state-of-the-art models without task repetition. When task boundaries are known, Query-Only Attention can also mitigate catastrophic forgetting. In addition, it achieves these benefits with lower computational cost compared to full attention.

Our analysis connects Query-Only models to MAML and full attention through the lens of global vs. local solutions and further relates them to k-nearest neighbors. We confirmed this relationship empirically across three benchmarks. Hessian-rank experiments support the role of curvature in sustaining plasticity across different approaches that mitigate loss of plasticity.

A key limitation is the reliance on a support set, which complicates mitigation of catastrophic forgetting. Future work will extend to larger and more diverse benchmarks, provide more rigorous theoretical analysis, and minimize reliance on explicit task support. From a responsible AI perspective, this limitation highlights the need for scalable, data-efficient continual learning mechanisms that preserve fairness and safety guarantees.

From a Responsible AI perspective, these results demonstrate that simplifying attention through meta-learning principles can yield reliable, transparent, and safe continual adaptation, supporting fairness, accountability, and trustworthiness in deployed foundation models.

Code Availability

The implementation of Query-Only Attention and all experiments are available at: https://github.com/gauthambekal93/query-only-attention-for-continual-learning.git.

Acknowledgments

We thank the organizers of the NeurIPS 2025 ResponsibleFM workshop for their feedback and support. We also appreciate the helpful discussions and suggestions from colleagues and peers who contributed to improving the quality of this work. We also thank anonymous reviewers for helpful feedback.

References

[1] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://proceedings.neurips.cc/paper/2023/file/8ed3d610ea4b68e7afb30ea7d01422c6-Paper-Conference.pdf.

[2] Qi Chen, Changjian Shui, Ligong Han, and Mario Marchand. On the stability-plasticity dilemma in continual meta-learning: Theory and algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/57587d8d6a7ede0e5302fc22d0878c53-Paper-Conference.pdf.

[3] Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 2024. doi: 10.1038/s41586-024-07711-7.

[4] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.64/.

[5] Mohamed Elsayed and A. Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8e5f0591943d8dae5702af12dcdcd2f6-Paper-Conference.pdf.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

[7] Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/file/f4dd765c12f2ef67f98f3558c282a9cd-Paper.pdf.

[8] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L. Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3390–3397. AAAI Press, 2018. URL https://dl.acm.org/doi/10.5555/3504035.3504450.

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/10.1073/pnas.1611835114.

[10] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of the ICML Deep Learning Workshop*, 2015. URL https://www.cs.utoronto.ca/~rsalakhu/papers/oneshot1.pdf.

[11] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. https://tinyimagenet.com/, 2015. Stanford CS231N Course Project.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2023. doi: 10.48550/arXiv.2312.00246. URL https://arxiv.org/abs/2312.00246.

[14] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 614–629. Springer, 2016. doi: 10.1007/978-3-319-46493-0_37. URL https://link.springer.com/chapter/10.1007/978-3-319-46493-0_37.

[15] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989. doi: 10.1016/S0079-7421(08)60536-8.

[16] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7308–7320, 2020. URL https://papers.neurips.cc/paper_files/paper/2020/file/518a38cc9a0173d0b2dc088166981cf8-Paper.pdf.

[17] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. URL https://arxiv.org/pdf/2209.11895.

[18] Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=LhY8QdUGSuw.

[19] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2007. URL https://www.eurasip.org/Proceedings/Eusipco/Eusipco2007/Papers/a5p-h05.pdf.

[20] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

[21] Jaehyeon Son, Soochan Lee, and Gunhee Kim. When meta-learning meets online and continual learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. doi: 10.1109/TPAMI.2023.3327373. URL https://ieeexplore.ieee.org/document/10684017.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[23] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

[24] Jiuqi Wang, Rohan Chandra, and Shangtong Zhang. Experience replay addresses loss of plasticity in continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2503.20018.

[25] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. doi: 10.1109/TPAMI.2024.3367329. URL https://doi.org/10.1109/TPAMI.2024.3367329.

[26] Shiguang Wu, Yaqing Wang, and Quanming Yao. Why in-context learning models are good few-shot learners? In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/pdf?id=iLUcsecZJp.

[27] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research (PMLR)*, pages 3987–3995, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

[28] Jianshu Zhang, Yankai Fu, Ziheng Peng, Dongyu Yao, and Kun He. Core: Mitigating catastrophic forgetting in continual learning through cognitive replay. *arXiv preprint arXiv:2402.01348*, 2024. doi: 10.48550/arXiv.2402.01348. URL https://arxiv.org/abs/2402.01348.

A Technical Appendices and Supplementary Material

A.1 Broader impacts

This work is primarily foundational research in continual learning. The proposed query-only attention mechanism and accompanying analysis aim to improve the understanding of plasticity and forgetting in neural networks. Potential positive impacts include enabling more efficient and adaptive AI systems, which could reduce retraining costs, improve energy efficiency, and support applications such as robotics, healthcare monitoring, and lifelong personal assistants.

At the same time, continual learning technologies can be misused in domains such as surveillance or profiling, where adaptive models might amplify privacy concerns or biases. While the present work is not directly deployable, these risks highlight the need for responsible use and safeguards in future applications.

Overall, this research contributes theoretical and empirical insights into the foundations of continual learning, with the aim of advancing the field in a transparent and beneficial direction.

A.2 Limitations

Our work has a few important limitations. First, the query-only attention model relies on a support set, which can be restrictive for mitigating catastrophic forgetting in practical continual learning scenarios. Second, our evaluation is limited to two benchmarks (Permuted MNIST and Slowly Changing Regression); broader validation on more complex datasets is needed to confirm generality. Third, while we provide theoretical analysis and intuition linking query-only attention to MAML and kNN, we do not include formal proofs. We acknowledge these as areas for future work, particularly in extending the experimental scope and strengthening the theoretical foundation.

A.3 Permuted MNIST Setup

The MNIST dataset [12] consists of 60,000 training and 10,000 test images of hand-written digits (0–9), each represented as a 28×28 grayscale image. To adapt this dataset for continual learning, we make the following modifications:

- **Train/test split.** For each task, we use the entire 60,000 original training images. These are further divided into 58,000 images for training and 2,000 images for evaluation within the task. The global MNIST test set is not used directly; instead, we re-sample 2,000 held-out

examples per task to serve as test data. This ensures consistency across tasks and keeps evaluation lightweight.

- **Downsampling.** To reduce computational cost, all images are downsampled from 28×28 to 7×7, giving 49 input features per image.

- **Task generation.** Each task corresponds to a new random permutation of the 49 input pixels. The same permutation is applied consistently to all 60,000 images within a task. Labels remain unchanged.

- **Continual stream.** The learner observes tasks sequentially. After completing all 58,000 training pairs of a given permutation, the learner encounters remaining 2000 data-points for testing, following which it encounters the next task (with a new permutation). In total, 7,000 such tasks are generated in the continual stream.

This setup forces the continual learner to adapt to a new input representation (permutation) at the start of each task, while retaining performance on past permutations. It is a widely used benchmark for evaluating both *plasticity* (ability to adapt to new tasks) and *stability* (ability to avoid catastrophic forgetting).

Table 1: Permuted MNIST configuration: Query-Only Attention (V1).

Setting	Value
Support size ($\|B\|$)	1000
Replay buffer size	`1000`
Distance metric	Learned Q_θ (query-only)
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0.0`
Batching	(batch size $= 400$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7500
Steps per task	`150`
(Input size, Hidden Size, Output Size, Hidden Layers)	(108, 100, 1, 9)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`6.5 hours/run`

Table 2: Permuted MNIST configuration: Query-Only Attention (V2).

Setting	Value
Support size ($\|B\|$)	200
Replay buffer size	`200`
Distance metric	Learned Q_θ (query-only)
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0.0`
Batching	batch size $= 400$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7500
Steps per task	`150`
(Input size, Hidden Size, Output Size, Hidden Layers)	(108, 100, 1, 9)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`4 hours/run`

Table 3: Permuted MNIST configuration: MAML.

Setting	Value
(Support size, query size, tasks per iteration) ($\|B\|$)	(10, 10, 5)
Replay buffer size	`50000`
Optimizer	Adam
Outer Learning rate	`1e-4`
Inner Learning rate	`1e-2`
Weight decay	`0.0`
Batching	batch size $= 400$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	75
Steps per task	`150`
(Input size, Hidden Size, Output Size, Hidden Layers) :	(49, 100, 10, 3)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`4.2 hours`

The attention network for continual learning is directly adapted from [24], which contains the comprehensive details.

Table 4: Permuted MNIST configuration: Attention Network baseline.

Setting	Value
Support size ($\|B\|$)	100
Attention	Full self-attention ($\mathcal{O}(n^2)$)
Replay buffer size	`100`
Optimizer	Adam
Learning rate	`5e-4`
Weight decay	`0.0`
Batching	batch size $= 400$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7500
Steps per task	`150`
Architecture	`10 layers, 1 head, d_model=59`
(Attention Layers, Attention Heads, Dimension) :	(10, 1, 59)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`10 hours/run`

The vanilla backpropagation and continual backpropagation algorithm is from paper [3], which contains more details.

Table 5: Permuted MNIST configuration: Vanilla Backpropagation.

Setting	Value
Support size	N/A
Replay buffer size	0 (no replay)
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0.0`
Batching	Online (batch size $= 1$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7500
Steps per task	`150`
(Input size, Hidden Size, Output Size, Hidden Layers) :	(49, 200, 10, 3)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`5.0 hours/run`

Table 6: Permuted MNIST configuration: Continuous Backpropagation.

Setting	Value
Support size	N/A
Replay buffer size	0 (no replay)
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0`
Batching	Online (batch size $= 1$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7500
Steps per task	`150`
(Input size, Hidden Size, Output Size, Hidden Layers) :	(49, 200, 10, 3)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`7.0 hours/run`

A.4 Split Image Net

We adopt the Split Image Net benchmark introduced by Dohare et al. [3]. However, due to computational constraints we utilize tiny version of the image net instead of full image net. We further downsample the images to 32 * 32 for faster computation. The remaining setup is same as in the original paper, ensuring comparability of results. For full details, we refer readers to the experimental protocol in [3]. Tiny image net consits of 200 labels and 500 images per label in training setup. The test setup consists of 50 labels per class. To incorporate tiny image net for continual learning, each task consits of randomly sampling images from 2 classes. Thus a task is a binary classification task with 1000 datapoints for training. At the end of training in that class we measure the accuray on 100 datapoints corresponding to 2 lables used in training. Since the training is purely online fashion a task is trained for a single epoch and then validation is carried out, followed by images for next task. All the below architectures presented use the same CNN architure as the starting point for input image transformation:

Table 7: Split Image Net: CNN architecture (same for all models)

Setting	Value
(Input channels, Output Channels,Kernel Size, Padding, MaxPool, Activation)	(3, 32, 5, 1, 2, Relu)
(Input channels, Output Channels,Kernel Size, Padding, MaxPool, Activation)	(32, 64, 3, 1, 2, Relu)
(Input channels, Output Channels,Kernel Size, Padding, MaxPool, Activation)	(64, 128, 3, 1, 2, Relu)
Init	Xavier uniform (weights), Zeros (biases)

Table 8: Split Image Net: Query-Only Attention.

Setting	Value
Support size ($\|B\|$)	180
Replay buffer size	200
Distance metric	Learned Q_θ (query-only)
Optimizer	Adam
Learning rate	1e-4
Weight decay	0.0
Batching	(batch size $= 10$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	9500
Steps per task	150
(Input size, Hidden Size, Output Size, Hidden Layers)	(2304, 128, 1, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	3 hours/run

Table 9: Split Image Net: MAML.

Setting	Value
(Support size, query size, tasks per iteration) ($\|B\|$)	(10, 10, 5)
Replay buffer size	20000
Optimizer	Adam
Outer Learning rate	1e-4
Inner Learning rate	1e-2
Weight decay	0.0
Batching	batch size $= 10$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	500
Steps per task	150
(Input size, Hidden Size, Output Size, Hidden Layers) :	(1152, 128, 2, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	5 hours (ran only first 100 tasks)

Table 10: Split Image Net: Attention Network baseline.

Setting	Value
Support size ($\|B\|$)	180
Attention	Full self-attention ($\mathcal{O}(n^2)$)
Replay buffer size	200
Optimizer	Adam
Learning rate	1e-4
Weight decay	0.0
Batching	batch size $= 10$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	9500
Steps per task	150
Architecture	3 layers, 1 head, d_model=130
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	4.5 hours/run

The vanilla backpropagation and continual backpropagation algorithm is from paper [3], which contains more details.

Table 11: Split Image Net: Vanilla Backpropagation.

Setting	Value
Support size	N/A
Replay buffer size	0 (no replay)
Optimizer	Adam
Learning rate	1e-4
Weight decay	0.0
Batching	Online (batch size $= 10$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	9500
Steps per task	150
(Input size, Hidden Size, Output Size, Hidden Layers) :	(1152, 128, 2, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	4.0 hours/run

Table 12: Split Image Net: Continuous Backpropagation.

Setting	Value
Support size	N/A
Replay buffer size	0 (no replay)
Optimizer	Adam
Learning rate	1e-4
Weight decay	0
Batching	Online (batch size $= 1$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7000
Steps per task	150
(Input size, Hidden Size, Output Size, Hidden Layers) :	(1152, 128, 2, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	5.5 hours/run

Table 13: EWC: Elastic Weight Consolidation.

Setting	Value
Lambda	100000
Optimizer	Adam
Learning rate	1e-4
Weight decay	0
Batching	Online (batch size $= 1$)
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	7000
Steps per task	150
(Input size, Hidden Size, Output Size, Hidden Layers) :	(1152, 128, 2, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	5.5 hours/run

A.5 Slowly changing regression (SCR)

We adopt the Slowly changing regression (SCR) benchmark introduced by Dohare et al. [3], which was designed to study loss of plasticity in continual learning. In this task, regression targets evolve gradually over time according to smoothly drifting functions, creating a non-stationary data stream. We use the same setup and data-generation procedure as in the original paper, ensuring comparability of results. For full details, we refer readers to the experimental protocol in [3].

Table 14: Slowly changing regression task: Query-Only Attention.

Setting	Value
Support size ($\|B\|$)	100
Replay buffer size	100
Distance metric	Learned Q_θ (query-only)
Optimizer	Adam
Learning rate	1e-4
Weight decay	0.0
Batching	batch size $= 1$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	800
Steps per task	10000
(Input size, Hidden Size, Output Size, Hidden Layers) :	(40, 20, 1, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	7 hours/run

Table 15: Slowly changing regression task: MAML.

Setting	Value
(Support size, query size, tasks per iteration) ($\|B\|$)	(10, 10, 5)
Replay buffer size	20000
Optimizer	Adam
Inner Learning rate	1e-2
Outer Learning rate	1e-4
Weight decay	0.0
Batching	batch size $= 1$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	800
Steps per task	10000
(Input size, Hidden Size, Output Size, Hidden Layers) :	(20, 40, 1, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	8.5 hour to run 100 tasks

Table 16: Slowly changing regression task: Attention Network.

Setting	Value
Support size ($\|B\|$)	100
Replay buffer size	100
Distance metric	Learned Q_θ (query-only)
Optimizer	Adam
Learning rate	1e-4
Weight decay	0.0
Batching	batch size $= 1$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	800
Steps per task	10000
(Attention Layers, Attention Heads, Dimension) :	(1, 1, 21)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	1$\times$ RTX 3090 24GB, CUDA 11.8
Wall-clock	14 hours/run

Table 17: Slowly changing regression task: Vanilla Backpropagation

Setting	Value
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0.0`
Batching	batch size $= 1$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	800
Steps per task	`10000`
(Input size, Hidden Size, Output Size, Hidden Layers) :	(20, 40, 1, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`8 hours/run`

Table 18: Slowly changing regression task: Continuous Backpropagation

Setting	Value
Optimizer	Adam
Learning rate	`1e-4`
Weight decay	`0.0`
Batching	batch size $= 1$
Seeds	20, 30, 40 (report mean $\pm$ std)
Tasks	800
Steps per task	`10000`
(Input size, Hidden Size, Output Size, Hidden Layers) :	(20, 40, 1, 1)
Init	Xavier uniform (weights), Zeros (biases)
Hardware	`1× RTX 3090 24GB, CUDA 11.8`
Wall-clock	`11 hours/run`

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 abstract and introduction clearly state that the paper introduces a query-only attention mechanism for continual learning, demonstrates its effectiveness in mitigating loss of plasticity and catastrophic forgetting, and provides supporting analysis via connections to K-nearest neighbor algorithm, attention network and MAML. The claims are consistent with both the theoretical discussion and the experimental evaluation (permuted MNIST and regression and SPLIT Image Net), and limitations are explicitly acknowledged. Please refer to section 6 and 7.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper explicitly acknowledges limitations, including the reliance on a support set (which may be restrictive in some continual learning scenarios), limited evaluation on permuted MNIST , Image Net and regression tasks, and the need for more rigorous theoretical analysis. We also note that computational cost and scalability of support-based methods remain open challenges for future work. Please refer to section 8 and A.2.

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: [No]

   Justification: The paper provides theoretical analysis and intuition (e.g., links between query-only attention, kNN, and MAML, as well as Hessian rank arguments), but does not contain formal theorems or complete proofs. The focus is on providing insight and empirical evidence rather than rigorous formalization. We acknowledge this as a limitation and identify formal proofs as an important direction for future work.

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: The paper specifies datasets (Permuted MNIST and Slowly Changing Regression and Split Image Net), model architectures (query-only attention, full-attention, BP, CBP, and MAML-style, EWC), training setup (online continual learning with single-pass data), evaluation metrics (forward and backward performance), and hyperparameters (support sizes, replay buffer sizes, etc.). Together, these details allow reproduction of the reported results. Code and implementation details (e.g., training loops, Hessian rank calculation) will be released to further ensure reproducibility. Please refer to appendix section A.3 and A.5 and A.4

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:We have provided link to GitHub URL for, ensuring reproducibility while preserving double-blind review. This will allow faithful reproduction of the main results. The GitHub is code currently being cleaned for easy reproducibility. Please refer to section A.3 , A.4 and A.5 for reproducibility.

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: The paper specifies datasets (Permuted-MNIST , Split Image Net and Slowly Changing Regression), evaluation protocols (forward vs. backward performance), optimizer (AdamW), replay buffer size, support set sizes, number of seeds (3), and training setup (single-pass, no epochs). Key hyperparameters are reported in the text, and additional details (exact learning rates, batch sizes, task counts) are provided in the appendix and GitHub code repository for reproducibility. Please refer to sections A.3 , A.4 and A.5.

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: [Yes]

   Justification: All experimental curves report the mean over 3 random seeds, with shaded regions showing ± 1 standard deviation across seeds. Please refer to 7, A.3, A.4 and A.5. This captures variability due to different random initializations and stochasticity in training.

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: We specify the compute resources used for all experiments. Runs were conducted on a single NVIDIA RTX 3090 GPU (24 GB VRAM) with 32 GB RAM. Each experiment (training across all tasks) required approximately 6–8 hours for permuted MNIST, 5-7 hours for split image net and 4–6 hours for slowly changing regression (SCR). The total reported results (3 seeds per experiment) were completed within 1 week. No large-scale distributed training or additional hidden compute was required beyond the reported experiments. Please refer to section A.3, A.4 and A.5.

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 work uses only publicly available datasets (MNIST, Tiny Image Net and synthetic regression benchmarks), contains no human or personally identifiable data, and does not raise safety, fairness, or environmental risks beyond typical compute usage for deep learning research. All experiments comply with the NeurIPS Code of Ethics.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This paper is primarily foundational research in continual learning and does not target immediate deployment. The potential positive impact lies in improving machine learning systems' ability to adapt over time without retraining, which could reduce computational costs and enable more sustainable, adaptive AI in areas such as robotics, healthcare monitoring, and personalized systems.

    Possible negative impacts include misuse in surveillance or adaptive profiling systems, where continual learning might enable intrusive or biased tracking over time. However, since our work focuses on simplified models (query-only attention) and theoretical analysis, these risks are indirect.

We explicitly acknowledge these as broader considerations, and stress that the current work is a step toward understanding fundamental mechanisms (plasticity vs. forgetting), not a directly deployable system. Please refer to section A.1

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: [Yes]

    Justification: This work is foundational research on continual learning algorithms and does not involve sensitive data or direct deployment. Potential positive impacts include advancing more adaptive and efficient machine learning systems, which could benefit areas such as robotics, personalized assistants, or scientific discovery. Potential negative impacts are indirect: continual learning techniques could be misused in surveillance, autonomous weapons, or other harmful applications if combined with inappropriate datasets or objectives. While the current work is limited to controlled benchmarks, we acknowledge these risks and encourage responsible use of such methods. Overall, the societal impact is primarily positive in advancing fundamental ML understanding.

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: We use only publicly available benchmark datasets and assets, including MNIST [12] (permuted variant), tiny image net and synthetic regression tasks (SCR). MNIST is distributed under a permissive license for research use, and we cite the original source. We also cite all baseline algorithms (e.g. transformers [22], CBP [3]). No proprietary or restricted data, models, or code were used, and all assets are properly credited and used within their intended terms.

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 work does not introduce new datasets or external assets. The only new contribution is the proposed query-only attention model, which is described in full detail in the paper and can be reproduced from the provided code. No human subjects or sensitive data are involved.

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 work does not involve crowdsourcing or research with human subjects. All experiments are purely computational, conducted on standard machine learning benchmarks (Permuted MNIST and SCR).

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 work does not involve human subjects, crowdsourcing, or any study requiring IRB approval. All experiments are computational and use standard public machine learning datasets.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No large language models (LLMs) were used as part of the core methodology or experiments. LLMs were only used as a general writing and editing assistant, which does not affect the scientific rigor or originality of the research.