Partially Shared Query-Key for Lightweight Language Models

Kai Yang
McGill University
kai.yang2@mail.mcgill.ca

Vahid Partovi Nia
Huawei Noah's Ark Lab
vahid.partovinia@huawei.com

Boxing Chen
Huawei Noah's Ark Lab
boxing.chen@huawei.com

Masoud Asgharian
McGill University
masoud.asgharian2@mcgill.ca

Abstract

Lightweight language models, such as *TinyBERT* 14.5M, have emerged as a critical area of research because of their implementation on resource-constrained hardware. These transformer models include significantly smaller parameter size, reduced memory and computational requirements. These features make such models highly suitable for deployment on small devices. We explore the concept of parameter sharing between the key and query weight matrices of a transformer model. The full query-key sharing which has already been proposed in the literature introduces a fully-quadratic attention matrix, oversimplifies directional dependencies and degrades pre-training loss. In contrast, *partial* parameter sharing balances complexity reduction and performance retention. Partial parameter sharing effectively addresses over-fitting while maintaining strong performance even with a high degree of shared parameters up to 95%. This provides a promising strategy for enhancing language models, specifically targeting small models.

1 Introduction

In recent years, lightweight language models, such as *TinyBERT*, have emerged as a critical area of research due to their efficiency in resource-constrained environments [Lan et al., 2019, Jiao et al., 2019, Radford et al., 2019, Sun et al., 2020, Clark et al., 2020, Iandola et al., 2020, Raffel et al., 2019]. While large language models (LLMs) have achieved remarkable success in various natural language processing (NLP) tasks, their substantial memory and computational requirements pose challenges for deployment on small devices. In contrast, lightweight models significantly reduce memory usage and computational demands, making them ideal for applications on mobile and edge devices. Furthermore, these models demonstrate greater robustness when trained on limited data, mitigating overfitting issues that often arise with larger models.

At the core of modern language models is the Transformer architecture, which utilizes the attention mechanism to capture dependencies between tokens by focusing on different parts of the input sequence [Vaswani et al., 2017]. Importantly, compared to traditional recurrent neural networks (RNNs), which process tokens sequentially, lightweight language models based on Transformers can leverage parallel processing on small devices. This parallelism enables more efficient computation, drastically reducing inference time, and making Transformer-based models far more suitable than RNNs for real-time applications on resource-constrained hardware.

A promising method to further improve lightweight models is through parameter sharing between the query and key components of the attention mechanism. Transformer variants such as *Reformer* [Kitaev et al., 2020] and *hyper-attention* [Han et al., 2023] have explored full parameter sharing,

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

showing that it can reduce memory usage without significant performance degradation. This suggests maintaining distinct weight matrices for query and key contributes to *over-parameterization*, while sharing these parameters leads to more efficient models by reducing the parameter count without compromising performance-crucial for small devices. Furthermore, this approach helps mitigate the risk of overfitting, particularly while training on limited data.

Despite the advantages of full parameter sharing, it is important to consider the role of distinct query and key vectors in creating an asymmetric attention matrix. The fully-shared key and query imposes $\mathbf{A}^\top = \mathbf{A}$ where $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$. Here we explore *to what extent the query and key can be treated as different matrices* while retaining the quality of the generated sentence.

Full parameter sharing between query and key does not fully capture the complexity of natural language, where word order is crucial. For example, the phrase "neural network" has a coherent meaning, but reversing the order to "network neural" disrupts this meaning entirely. The asymmetry introduced by distinct query and key vectors is essential for capturing directional dependencies in natural language, which is why reducing this asymmetry may have performance implications.

We investigate the impact of *query-key parameter sharing* within the attention mechanism, specifically focusing on lightweight language models. We hypothesize that *partial parameter sharing* reduces the complexity of the model, while maintaining the benefits of an asymmetric attention matrix. We found that partial parameter sharing not only addresses overfitting, but also preserves pre-training and validation loss.

Through this exploration of parameter sharing, we highlight lightweight Transformer models for deployment in resource-constrained environments. By balancing complexity reduction without compromising performance, this study provides insight into designing more efficient models capable of capturing token dependencies in natural language.

2 Methodology

To investigate the effects of parameter sharing between the query and key vectors, we introduce a method that constructs a weight matrix for a shared component. Let

$$\mathbf{W}_Q \coloneqq [\mathbf{W} \quad \mathbf{W}_{Q'}] \tag{1}$$

$$\mathbf{W}_K \coloneqq [\mathbf{W} \quad \mathbf{W}_{K'}] \tag{2}$$

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q = [\mathbf{X}\mathbf{W} \quad \mathbf{X}\mathbf{W}_{Q'}]$$

$$\mathbf{K} = \mathbf{X}\mathbf{W}_K = [\mathbf{X}\mathbf{W} \quad \mathbf{X}\mathbf{W}_{K'}]$$

$$\mathbf{Q}\mathbf{K}^\top = \mathbf{X}\left(\mathbf{W}\mathbf{W}^\top + \mathbf{W}_{Q'}\mathbf{W}_{K'}^\top\right)\mathbf{X}^\top. \tag{3}$$

The shared weight matrix $\mathbf{W}$ is integrated with separate weight matrices for the distinct components of the query and key (i.e., $\mathbf{W}_{Q'}$ and $\mathbf{W}_{K'}$ in (1), (2), hence leading to partial parameter sharing. The resulting vectors are then concatenated, enabling the model to exploit shared information while preserving unique features for each vector. Furthermore, by reducing the dimensionality of the query and the key weight parameters, this method simplifies the model architecture, achieving greater efficiency without compromising the performance (see Figure 1). Note that $\mathbf{W}\mathbf{W}^\top + \mathbf{W}_{Q'}\mathbf{W}_{K'}^\top$ in (3) can be interpreted as a low-rank approximation $\mathbf{W}_{Q'}\mathbf{W}_{K'}^\top$ applied to a low-rank query and key matrix with total sharing $\mathbf{W}\mathbf{W}^\top$. Previous studies have demonstrated the effectiveness of LoRA in preserving model performance while significantly reducing computational costs, particularly during different stages of fine-tuning (see, for example, [Hu et al., 2021, Wang et al., 2024, Xu et al., 2023, Zeng and Lee, 2023]).

We implemented this method in two popular toy experimental settings, specifically, Tiny Shakespeare [Karpathy, 2015] and the Oxford American Essays dataset [Doshi, 2022]. Both datasets, composed of character-level text data, were divided into training and validation sets with a $90/10$ split, with 10% of the data used for validation. Each character in the text was tokenized as an integer that represents its unique occurrence in the dataset. The model processes sequences of these tokens, predicting the next character in the sequence, with the input data offset by one character.

Our model utilizes a multihead attention mechanism with 6 attention heads and 6 Transformer layers, each comprising an attention block followed by a feedforward neural network (FNN). The embedding

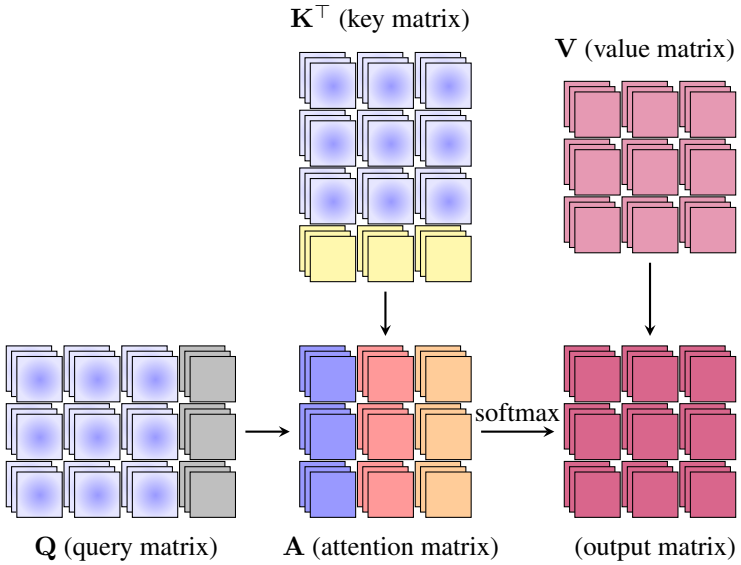

Figure 1: The concept of partial-sharing between the query and key. The radial blue area represents the submatrix where the query and key share the same values for each token.

dimensionality is set to 384 to balance model capacity and computational efficiency. We pre-trained the model with a batch size of 64 and a sequence length (block size) of 256 characters, which allowed the model to capture long-range dependencies while managing computational load. The learning rate was set to 3×10^{-4}, chosen to ensure stable convergence during pre-training. A dropout rate of 20% was applied within the FNN layers to mitigate overfitting. Training was carried out over $5K$ iterations, with the cross-entropy objective function evaluated every 50 iterations. The model's performance on both the training and validation sets was regularly evaluated by averaging the loss over 200 batches, providing a robust measure of generalization.

The implementation was straightforward, requiring only minor modifications to the existing attention mechanism and FNN in Transformer models, specifically based on the nanoGPT model and code from Andrej Karpathy [Karpathy, 2023]. These modifications allowed us to assess the benefits of shared parameters in a controlled, reproducible environment, offering valuable insights into the trade-offs between model complexity and performance.

3 Experiments

In our experiments, we analyzed the impact of varying the degree of parameter sharing between the query and the key on the model performance. We evaluated our models on the Tiny Shakespeare and Oxford American Essays datasets, focusing on the cross-entropy loss during both training and validation.

3.1 Impact of Full Parameter Sharing

As depicted in Figures 2 and 3, fully sharing the query and key vectors consistently resulted in a higher cross-entropy loss during both training and validation compared to other configurations — this degradation is particularly evident in the pretraining loss. This performance degradation can be attributed to the enforced symmetry in the attention score matrix, which oversimplifies the attention mechanism. In natural language processing, directional relationships between tokens are crucial; when these relationships are treated symmetrically, as in the case of fully shared query and key vectors, the model loses the ability to capture important linguistic nuances, leading to reduced effectiveness.

3.2 Partial Sharing

Interestingly, models with partial sharing of query and key parameters performed similarly to models in which the query and key were kept entirely separate. Furthermore, we pushed the proportion of sharing close to 1, with proportions such as 90% and 95%, and found that the performance remained comparable to the original setting with fully separate query and key vectors. This was consistent between training and validation, as shown in Figures 2 and 3.

Moreover, we observed that overfitting effects after a certain number of `AdamW` iterations, specifically between $3.5K$ and $4K$ iterations. This suggests that partial sharing of the query and key vectors is, in fact, beneficial, as it can help mitigate overfitting by reducing the model's complexity while still maintaining a strong performance.

The results suggest that *partial* sharing strikes an effective balance: it reduces the model complexity while preserving the ability to capture the directional dependency structure inherent in the data. This balance is maintained even when the shared proportion is high, demonstrating that partial sharing does not significantly sacrifice performance, even when pushed close to complete sharing.

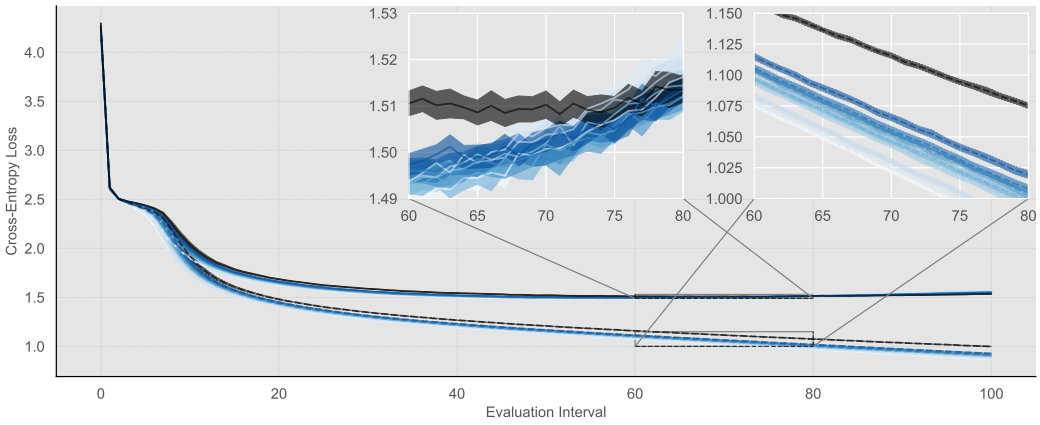

Figure 2: Cross-entropy *pretraining* (dashed lines) and *validation* (solid lines) loss on the Tiny Shakespeare dataset, displaying the mean and 95% confidence intervals over 30 simulation runs with varying levels of shared query-key proportions. Each evaluation interval represents 50 optimization steps. Increasing shades of blue represent higher levels of sharing, while black indicates full sharing.

4 Conclusion

In this work, we investigated the impact of partial parameter sharing between query and key components in lightweight Transformer models. Our experiments on the Tiny Shakespeare dataset reveal that while fully shared query and key vectors tend to oversimplify the attention mechanism and degrade performance, partial sharing effectively balances model complexity and performance. Specifically, even with a high degree of sharing (e.g., 90% and 95%), the performance remains comparable to that of fully separate query and key vectors. This indicates that partial parameter sharing can reduce computational demands without significantly compromising the model's effectiveness.

These findings underscore the viability of partial sharing as a strategy to optimize lightweight language models, maintaining crucial directional relationships in the data. Future research could explore varying the degree of parameter sharing across different Transformer layers and assess its impact on a range of NLP tasks. This approach promises to be a valuable direction for enhancing the efficiency of models in resource-constrained environments.

References

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. March 2020. doi: 10.48550/ARXIV.2003.10555.

Ketan Doshi. Oxford american essays dataset. https://huggingface.co/datasets/iamketan25/essay-instructions-dataset, 2022. URL https://huggingface.co/datasets/iamketan25/essay-instructions-dataset. Available at.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P. Woodruff, and Amir Zandieh. Hyperattention: Long-context attention in near-linear time. October 2023. doi: 10.48550/ARXIV.2310.05869.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. June 2021. doi: 10.48550/ARXIV.2106.09685.

Forrest N. Iandola, Albert E. Shaw, Ravi Krishna, and Kurt W. Keutzer. Squeezebert: What can computer vision teach nlp about efficient neural networks? June 2020. doi: 10.48550/ARXIV.2006.11316.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. September 2019. doi: 10.48550/ARXIV.1909.10351.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015. URL http://karpathy.github.io/2015/05/21/rnn-effectiveness/. Available at.

Andrej Karpathy. nanogpt video lecture series, 2023. URL https://github.com/karpathy/ng-video-lecture.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. 2020. doi: 10.48550/ARXIV.2001.04451.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. September 2019. doi: 10.48550/ARXIV.1909.11942.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. October 2019. doi: 10.48550/ARXIV.1910.10683.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. April 2020. doi: 10.48550/ARXIV.2004.02984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. doi: 10.48550/ARXIV.1706.03762.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. July 2024. doi: 10.48550/ARXIV.2407.05000.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. September 2023. doi: 10.48550/ARXIV.2309.14717.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. October 2023. doi: 10.48550/ARXIV.2310.17513.

A Appendix / Supplemental Material

For additional experimentation, we used the Oxford American Essays dataset [Doshi, 2022], which consists of character-level text data from a collection of American essays. The dataset was split into training and validation sets using a 90/10 ratio. Each character was tokenized similarly to the Tiny Shakespeare dataset, representing unique occurrences.

All model architecture, hyperparameters, and training settings for the Oxford American Essays dataset were identical to those used for the Tiny Shakespeare dataset. The same approach to partial parameter sharing between the query and key matrices, as described in the main paper, was applied.

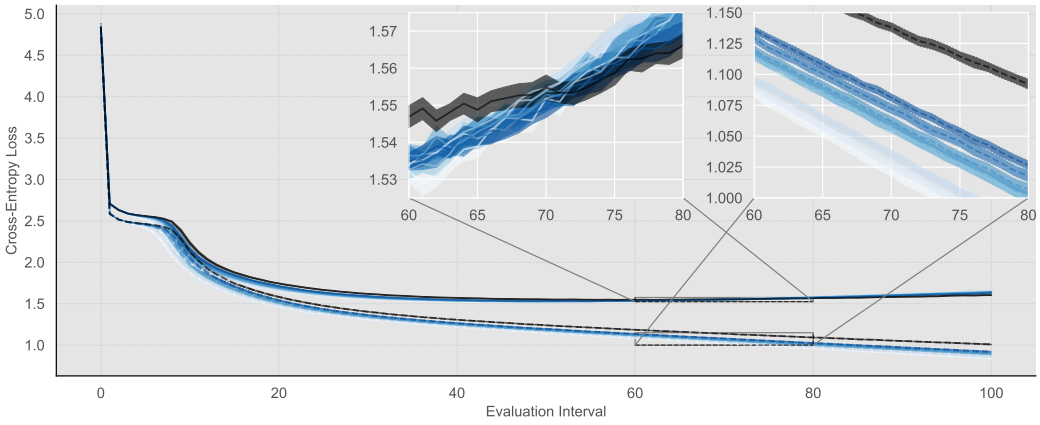

Figure 3: Cross-entropy *pretraining* (dashed lines) and *validation* (solid lines) loss on the Oxford American Essays dataset, displaying the mean and 95% confidence intervals over 30 simulation runs with varying levels of shared query-key proportions. Each evaluation interval represents 50 optimization steps. Increasing shades of blue represent higher levels of sharing, while black indicates full sharing.