Projected Compression: Trainable Projection for Efficient Transformer Compression

Maciej Stefaniak*, Michał Krutul*, Jan Małaśnicki, Maciej Pióro, Jakub Krajewski, Sebastian Jaszczur, Marek Cygan, Kamil Adamczewski, Jan Ludziejewski

University of Warsaw, IDEAS NCBR, Wrocław University of Science and Technology

TL;DR

We introduce **Projected Compression (PC)**, a method for reducing Transformer model size using trainable projections. It preserves access to original weights during training and merges them into a compact form **without increasing per-token FLOPs**. The approach **outperforms hard pruning with retraining (HPR)**, especially at high compression rates.

Projected Compression

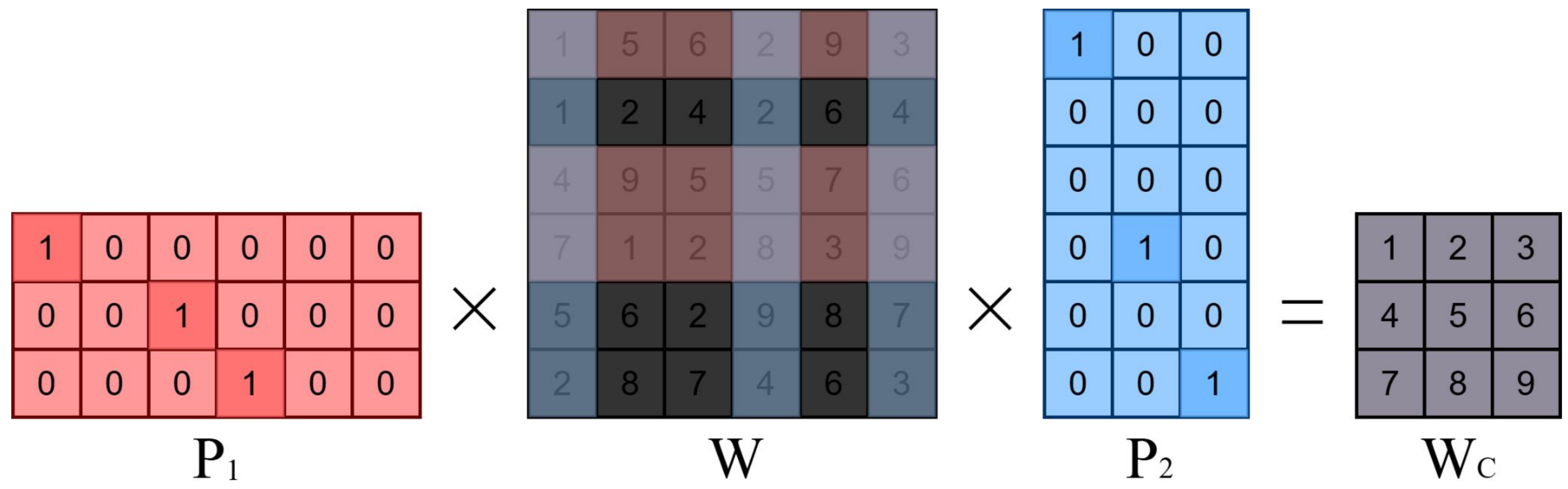

Figure 1: Illustration of a PC module, where **P_1** and **P_2** are **projection matrices**, **W** is a **frozen base model parameters** and **W_C** is compressed weights matrix.

Training Loss

Method	1B	2B	4B	6B
	Tokens processed during training			
90% compression				
HPR	3.3870	3.2198	3.1032	3.0500
PC	**3.3155**	**3.1787**	**3.0809**	**3.0366**
70% compression				
HPR	2.9542	2.8546	2.7767	2.7370
PC	**2.9387**	**2.8439**	**2.7700**	**2.7310**
50% compression				
HPR	**2.7960**	**2.7005**	**2.6451**	**2.6134**
PC	2.7986	2.7038	2.6476	2.6154
50% compression with dataset alignment				
HPR	**2.7644**	**2.6966**	2.6495	2.6223
PC	2.7656	2.6978	**2.6493**	**2.6216**

Table 1: **PC vs HPR** - cross-entropy **loss** of **Llama3 1B compressed** model. **Best** loss from each method pair **is bolded**.

Cost-free projections optimization

Forward and backward passes over tokens run **through the compressed matrix WC, not the P1, W and P2**. Gradients accumulate in the small parameter space of WC, then are used to calculate gradients for P1, P2 via W .

There are two costs for each training step:

- **Fixed costs** that do not scale with number of tokens in batch: building WC = P1W P2, calculating ∇P1 , ∇P2 from ∇WC.
- **Costs that scales with the number of tokens processed in the batch**: forward and backward cost through WC to calculate ∇WC which (step).

With sufficiently **large batch size**, projection optimization costs becomes negligible. Consequently, the total **cost per training step** is **nearly equivalent to** that of retraining **a** compressed **transformer model** obtained by hard pruning.

It should be noted that PC comes at the expense of additional memory usage that can be mitigated with CPU RAM offloading.

Downstream tasks

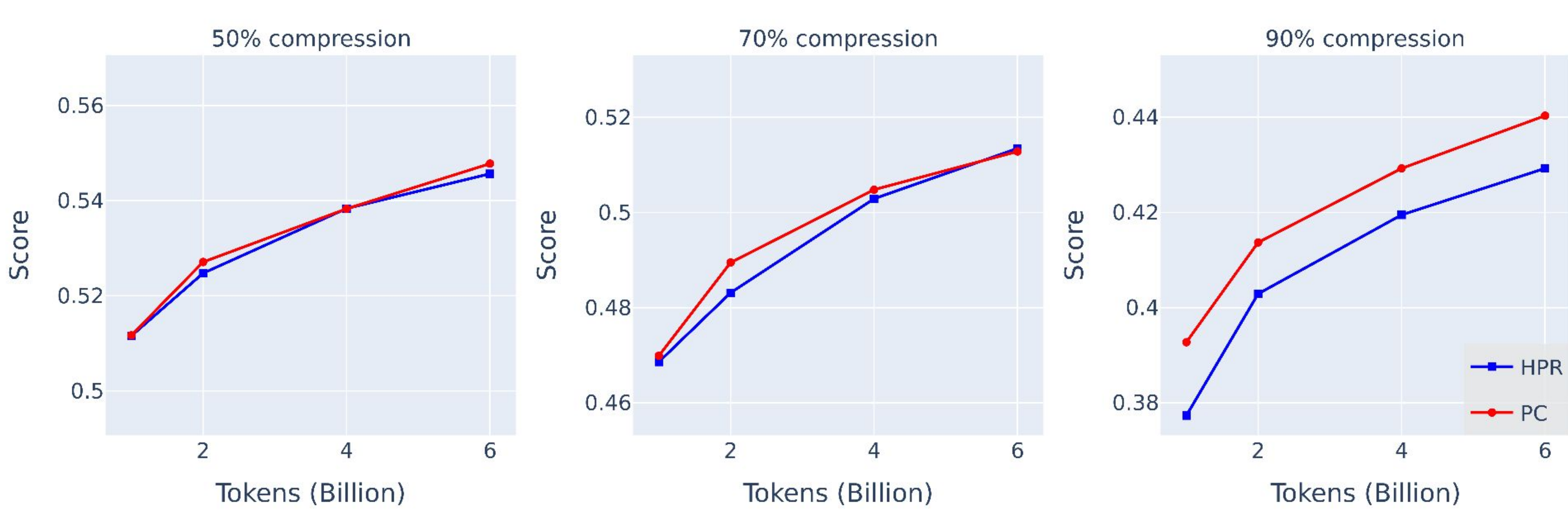

Figure 2: **PC vs HPR** - Averaged **benchmarks** for **compressed Llama3 1B** model. Each data point is obtained from a separate training run.

Full Paper

LLMRandom Nano Framework