DOTRESIZE: Reducing LLM Width via Discrete Optimal Transport-based Neuron Merging

Neha Verma [1] **Kenton Murray** [1 2] **Kevin Duh** [1 2]

Abstract

Structured pruning methods designed for Large Language Models (LLMs) generally focus on identifying and removing the least important components to reduce model size. In this work, we question the need for total removal, instead exploring how to integrate information from less important structures back into the compressed model. We specifically focus on neuron width pruning by framing it as a **D**iscrete **O**ptimal **T**ransport problem, leading to our proposed method DOTRESIZE, a novel Transformer compression method that uses optimal transport theory to transform and compress model width. To ensure applicability within the Transformer architecture, we motivate and incorporate necessary entropic regularization and matrix factorization techniques into the transportation maps produced by our method. Unlike pruning-based approaches which discard neurons based on importance measures, DOTRESIZE reprojects the entire neuron width, allowing the retention and redistribution of useful signal across the reduced layer. Empirical results show that DOTRESIZE serves as a useful add-on to pruning techniques, while achieving measurable reductions in real-world computational cost.

1. Introduction

Structured pruning is a popular approach to model compression because it directly reduces the on-disk size of LLMs and accelerates inference, without requiring specialized hardware or libraries often needed for unstructured pruning. One variety of structured pruning reduces the width of weights across model layers, resulting in compressed models with more computational throughput (Ma et al., 2023; Ashkboos et al., 2024a). In general, these width pruning techniques aim to identify and discard neurons within each layer that are deemed less important. However, total removal may risk discarding undervalued but important signal and lead to suboptimal compression.

Recent work has exploited structural invariances within Transformers to enable model merging, activation quantization methods, and new pruning techniques (Singh & Jaggi, 2020; Ainsworth et al., 2023; Yang et al., 2024b; Theus et al., 2024). These methods apply transformations π to parameters θ while strictly or approximately preserving the functionality of the model, i.e., $f(x|\theta) \approx f(x|\pi(\theta))$. Relevant examples include QuaRot and SliceGPT, where rotation matrices are applied to Transformer weights to prepare for quantization or pruning, respectively (Ashkboos et al., 2024b;a). We extend this invariance-based paradigm include optimal transport maps, allowing us to achieve the recombination principle we seek to employ.

In this work, we move beyond pruning's binary choice of whether to keep or discard neurons by introducing a new, principled approach to merging-based LLM compression based on discrete optimal transport. This approach represents a first step toward unifying model merging and structured pruning, aiming to reincorporate the "unaccounted" signal that is typically lost during compression. Although a prior transport-based approach has been proposed for ResNet image models (Theus et al., 2024), this approach does not apply to LLM architectures. Our method does not require recovery fine-tuning, offering a promising new direction for efficient, training-free model compression. When applied atop gradient-free LLM neuron pruning methods, DOTRESIZE frequently recovers additional performance by re-incorporating signal from neurons designated for pruning. Our method is summarized in Figure 1.

Our contributions are three-fold: (i) We propose a training-free neuron width reduction method in LLMs that uses discrete optimal transport to recombine signal from neurons that would otherwise be pruned. (ii) To apply our transport-based maps, we re-examine a specific invariance result found in Ashkboos et al. (2024a) that enables the appli-

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD, USA. Correspondence to: Neha Verma <nverma7@jhu.edu>.

Workshop on Weight-Space Symmetries, held in conjunction with the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

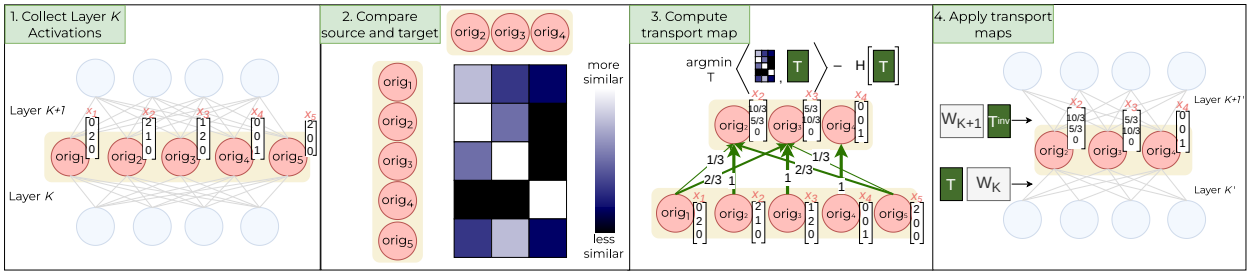

Figure 1. A depiction of our neuron width merging strategy. In panel 1, we collect activations from layer K. In 2, we select a subset of neurons from this layer and compute pairwise similarities between activations of layer K and the subset. In 3, we compute the optimal transport map T by optimizing this map according to the similarities and entropic regularization. Finally, we demonstrate replacing layer K with the subset of neurons, after transforming its weights with T and layer $K+1$'s weights with T^{inv}, resulting in width reduction.

cation of orthogonal maps throughout LLM weights. We extend this set of invariant maps to any invertible map, via the application of QR-decomposition to respect the orthogonal-only invariance of RMSNorm. (iii) We show that across language modeling and zero-shot LLM tasks, our method can frequently regain some of the performance loss caused by neuron-level pruning in directly comparable methods, supporting the core motivation of our work.

2. DOTRESIZE

Our DOTRESIZE width reduction method draws on entropy-regularized discrete optimal transport to map a layer's full neuron distribution onto a representative subset, guided by activation signatures over example data.

Computing the transport maps Given original model width d_{o} and new model width d_{n}, we follow Singh & Jaggi (2020) by defining our source and target distributions as uniform: $\mathbf{a} = \mathbf{1}_{d_{\text{o}}}/d_{\text{o}}, \mathbf{b} = \mathbf{1}_{d_{\text{n}}}/d_{\text{n}}$. We select $d_{\text{n}} < d_{\text{o}}$ neurons according to an underlying pruning strategy. We define the cost matrix of transporting mass from $\mathbf{a}$ to $\mathbf{b}$ as the pairwise distances between the activations of the d_{o} source neurons and the d_{n} selected neurons, across k exemplar tokens. In our work, we use ℓ_1 norm to compute pairwise distances. This results in cost matrix $C \in \mathbb{R}^{d_{\text{o}} \times d_{\text{n}}}$, where for activations $x_i, x_j \in \mathbb{R}^k$ corresponding to source neuron i and target neuron j, $C_{ij} = ||x_i - x_j||_1$.

We compute transport maps using an entropy-regularized objective. Sinkhorn regularization improves computational tractability and promotes a smooth mapping where each target neuron represents a combination of source neurons (Cuturi, 2013; Imfeld et al., 2024). For the Sinkhorn objective, the larger the parameter λ, the "softer" the transportation map is due to the flattening effect of the entropy term.

$$T = \arg\min_{T} \langle T, C \rangle - \lambda H(T) \text{ s. t. } T\mathbf{1}_{d_{\text{o}}} = \mathbf{a}, T^T\mathbf{1}_{d_{\text{n}}} = \mathbf{b}$$

After applying map T to the one or more weight matrices that produced the activations used for this optimization, we can apply an inverse transformation to the following weight matrix such that functional invariance is accounted for.

Extension to invertible maps Applying general invertible transportation matrices to Transformer weights will not work due to their interaction with RMSNorm. In Ashkboos et al. (2024a), orthogonal matrices were used as they are norm preserving. However, we are able to relax this constraint to general invertible transportation matrices to accommodate our transport maps. While invertible matrices can alter vector norms, we circumvent this by using a QR decomposition step $T = QR$ such that the Q matrix is applied to the input vector before the RMSNorm layer, and the R matrix is applied after RMSNorm application. Given decomposition $T = QR$, inverse matrix T^{-1} and data vector $\mathbf{x}$, we have the following relationship:

$$\text{RMS}(\mathbf{x}Q)RT^{-1} = \frac{\mathbf{x}Q}{\|\mathbf{x}Q\|}RT^{-1} = \frac{\mathbf{x}QRT^{-1}}{\|\mathbf{x}\|} = \text{RMS}(\mathbf{x})$$

A simple explanation of steps in this equality appears in Appendix B. Given the associativity of matrix multiplication, we are able to QR-decompose our computed transport map and then store just $M = Q$ and $M^{\text{inv}} = RT^{-1}$ as our final transport and untransport maps. An overview of the application of these maps is in Figure 2.

Applying maps We apply two map pairs per layer to dense, SwiGLU transformers. $\{M_A, M_A^{\text{inv}}\}$ target W_o from attention, and $\{W_{\text{gate}}, W_{\text{up}}\}$ in the FFN. $\{M_F, M_F^{\text{inv}}\}$ target W_{down} in the FFN, and $\{W_K, W_Q, W_V\}$ in attention. All maps are absorbed into neighboring weight matrices. Full application details are in Appendix D.

3. Experimental Settings

Models, Data, and Evalution We use 5 LLMs in our study: Llama 3.1 at 8B and 70B parameters (Grattafiori et al., 2024), Mistral at 7B (v0.3) and 12B (NeMo) parameters (Yang et al., 2024a), and Phi-4 at 14B parameters (Abdin et al., 2024). For testing their capa-

Width Reduction	Method	Llama-3.1		Mistral		Phi-4
		8B	70B	7B	12B	14B
0%	-	6.24	2.81	5.32	5.74	6.46
20%	Magnitude Prune	29.33	12.72	15.68	36.78	14.07
	+DOTRESIZE	**16.57**	**9.74**	**10.08**	**16.78**	**11.07**
30%	Magnitude Prune	108.23	86.48	43.42	128.04	66.48
	+DOTRESIZE	**36.20**	**32.48**	**20.26**	**32.15**	**29.79**
20%	SliceGPT	11.25	6.92	**6.76**	8.11	7.67
	+DOTRESIZE	**10.24**	7.01	6.98	**7.96**	7.71
30%	SliceGPT	19.42	9.30	**8.69**	10.73	9.14
	+DOTRESIZE	**16.46**	9.26	8.89	10.68	9.20

Table 1. Perplexity performance of models on Wikitext-2 under different width reduction strategies and different levels of width reduction. Perplexity results that reflect > 0.1 PPL reduction are bolded. In comparing pruning to DOTRESIZE, we frequently see an improvement in perplexity, indicating that using optimal transport to reduce neuron width can be superior to solely pruning neurons.

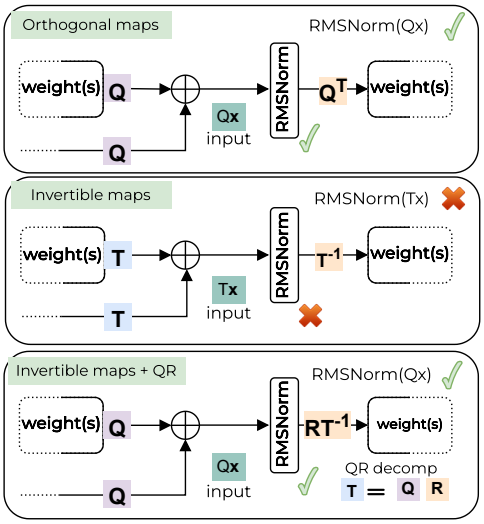

Figure 2. QR-decomposition enables applying general invertible matrices to Transformer weight matrices without perturbing RMSNorm behavior. By decomposing $T = QR$ and absorbing R into the T^{-1} inverse, the resulting orthogonal component preserves RMSNorm invariance in pre-norm architectures.

bilities before and after compression, we compute perplexity on the English Wikitext-2 dataset (Merity et al., 2017). In addition, we use 5 benchmarks sourced from `lm-evaluation-harness` (Gao et al., 2024a): ARC-Challenge and ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and Winogrande (Sakaguchi et al., 2021). In all language modeling experiments, we use a small subset (128 samples) of the Wikitext-2 training data to compute our transportation maps. Samples are length 2048. For all zero-shot experiments, we use the same number of tokens from the Stanford Alpaca dataset for instruction following (Taori et al., 2023).

Pruning methods We apply DOTRESIZE atop two training-free neuron pruning methods. We use magnitude pruning as our first, where at each layer, we remove the neurons with lowest average ℓ_2 activation norm. Pruning maps are propagated to the residual connections to allow unique maps per layer. The second pruning method is SliceGPT, which first applies PCA to reproject features, and then prunes neurons with least variance (Ashkboos et al., 2024a). We directly compare our method to SliceGPT by reprojecting activations by their PCA basis before applying DOTRESIZE to reduce width. Neurons are selected by l_1 norm. We differ from their method by forgoing their slicing matrix and instead using our transport map in its place. We use Sinkhorn regularization $\lambda = 0.1$ for both settings.

4. Results

We demonstrate the performance of pruning methods with and without DOTRESIZE on language modeling and zero-shot tasks. Additional analysis of our method appears in Appendix G.

4.1. Language modeling

Table 1 presents language modeling results for the uncompressed models, existing pruning techniques, and our method applied atop the pruning methods. In applying DOTRESIZE atop Magnitude Prune, DOTRESIZE substantially lowers model perplexity by preserving and leveraging signal from neurons that would otherwise be discarded. While Magnitude Prune only retains the neurons with the highest ℓ_2 activation norms, DOTRESIZE uses these high-norm neurons as a support set and redistributes the full neuron width onto them based on similarity. Although this redistribution can sometimes reduce the influence of neu-

Width Reduction	Method	Llama-3.1-8B	Llama-3.1-70B	Mistral-7B	Mistral-12B	Phi-4 14B
0%	-	73.65	80.11	73.39	75.66	73.76
20%	SliceGPT	60.74	**71.48**	62.89	51.43	71.04
	+DOTRESIZE	**60.94**	67.43	**63.14**	**63.14**	**72.08**
30%	SliceGPT	51.89	**67.09**	**54.96**	45.94	**66.84**
	+DOTRESIZE	**52.65**	65.00	53.67	**53.71**	66.44

Table 2. Average zero-shot accuracy (%) across ARC-C, ARC-E, HellaSwag, PIQA, and Winogrande under different width reduction settings. Bolded results reflect the higher accuracy within each model/sparsity comparison. Full per-task results in Appendix E.

rons that would have been retained by Magnitude Prune, the advantage of preserving signal from the pruned neurons more than compensates for this effect, as seen by the perplexity reduction.

In comparing the DOTRESIZE results with SliceGPT on Llama-3.1-8B, our method reduces perplexity by almost 3 points at 30% width reduction, and over 1 point for 20% width reduction. While performance on Mistral-7B does not improve from using DOTRESIZE, other models either show improvement or parity. Given that DOTRESIZE can improve performance considering the differences in the objectives to reduce neuron width, we are able to demonstrate an interesting limitation of SliceGPT. While PCA is the ℓ_2-optimal strategy for linear dimensionality reduction, our results show that minimizing ℓ_2 activation distance does not always correlate with improved downstream performance. This result suggests that relying on ℓ_2 activation distance as a proxy for model performance, which is common in compression applications (Dong et al., 2017; Frantar & Alistarh, 2022), may miss other aspects of model behavior.

4.2. Zero-shot performance on downstream applications

Given that SliceGPT with and without DOTRESIZE outperforms Magnitude Pruning-based DOTRESIZE in language modeling, we evaluate only SliceGPT and +DOTRESIZE on zero-shot tasks. Average accuracies across the 5 tasks are in Table 2. As seen in the table, several model/sparsity combinations are improved using DOTRESIZE, including Mistral-Nemo 12B, which sees a substantial 11% average accuracy improvement across tasks at 20% sparsity. Additionally, using DOTRESIZE improves results for Phi-4 at 20%, which is remarkably robust to both our compression method and SliceGPT. At 20% sparsity, this model retains 98% of its original average zero-shot performance, and even sees higher accuracy on the ARC-Easy and Challenge sets after compression versus in the base model.

We also observe model-specific differences in both language modeling and zero-shot performance, with certain models being very receptive to this signal recombination approach, and others benefiting only slightly or not benefiting any. These variations potentially stem from training differences across LLMs, which can affect their compatibility with merging-based compression. To investigate this further, we plot the effective rank of activations (at 99% variance) for Llama-3.1-8B, which is enhanced by DOTRESIZE, and Mistral-7B, which follows SliceGPT-like trends. These results, captured during compression, are detailed in Appendix F. Figure 3 shows that Mistral activations demonstrate a much lower effective rank than Llama activations in earlier layers, suggesting that Mistral activations contain most of its signal within its first few neurons after PCA reprojection. Despite this concentration, we find our method remains sensitive to varying-width reduction strategies across layers, mirroring findings in Ashkboos et al. (2024a). These results suggest our method is best applied to models with high-rank activation signal, whereas for models with lower-rank, concentrated signal, PCA-based pruning methods remain an effective strategy.

5. Conclusion and future work

In this work, we introduce DOTRESIZE, a novel LLM compression method which frames neuron width reduction as an optimal transport problem to re-project a layer's neurons, specifically avoiding the pruning of any particular neuron. With this approach, we are able to demonstrate improved performance across several LLMs compared to pruning-based alternatives, demonstrating the importance of incorporating the full signal of a layer and the potential insufficiencies of pruning. We find some models to be markedly receptive to the addition of a merging-based objective in width reduction, with Phi-4 maintaining almost 98% of zero-shot performance on downstream tasks while improving on some individual tasks at 20% sparsity, and Mistral-Nemo achieving an over 10% gain in average accuracy across tasks over SliceGPT. Furthermore, we extend computational invariance results in Transformers by demonstrating that invertible matrices can be applied to Transformer weights via a QR-decomposition step without changing the model function. Model-specific variations we observe suggest that intrinsic model properties may influence OT-based compressibility; understanding these factors could enable more adaptive compression strategies. More broadly, discrete optimal transport offers a principled framework for parameter transformation that may extend beyond compression.

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Ainsworth, S., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

Ashkboos, S., Croci, M. L., do Nascimento, M. G., Hoefler, T., and Hensman, J. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*, 2024a.

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. Quarot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, 2024b.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Dalvi, F., Sajjad, H., Durrani, N., and Belinkov, Y. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4908–4926. Association for Computational Linguistics, 2020.

Dong, X., Chen, S., and Pan, S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in neural information processing systems*, 30, 2017.

Frantar, E. and Alistarh, D. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 2024a.

Gao, S., Lin, C.-H., Hua, T., Tang, Z., Shen, Y., Jin, H., and Hsu, Y.-C. Disp-llm: Dimension-independent structural pruning for large language models. *Advances in Neural Information Processing Systems*, 2024b.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Imfeld, M., Graldi, J., Giordano, M., Hofmann, T., Anagnostidis, S., and Singh, S. P. Transformer fusion with optimal transport. In *The Twelfth International Conference on Learning Representations*, 2024.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2, 1989.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. Spinquant: Llm quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720, 2023.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Nanda, V., Speicher, T., Dickerson, J., Gummadi, K., Feizi, S., and Weller, A. Diffused redundancy in pre-trained representations. *Advances in Neural Information Processing Systems*, 36:4055–4079, 2023.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Singh, S. P. and Jaggi, M. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

Srinivas, S. and Babu, R. V. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

Stoica, G., Bolya, D., Bjorner, J. B., Ramesh, P., Hearn, T., and Hoffman, J. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*, 2024.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.

Theus, A., Geimer, O., Wicke, F., Hofmann, T., Anagnostidis, S., and Singh, S. P. Towards meta-pruning via optimal transport. In *The Twelfth International Conference on Learning Representations*, 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Verma, N. and Elbayad, M. Merging text transformer models from different initializations. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Yang, E., Shen, L., Guo, G., Wang, X., Cao, X., Zhang, J., and Tao, D. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024b.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

A. Related work

Neuron-level pruning Unlike unstructured pruning, which lacks real-world latency and memory savings (LeCun et al., 1989), structured pruning can offer hardware-agnostic acceleration and real-world memory savings. Although converting unstructured pruning to semi-structured can improve latency (Frantar & Alistarh, 2023; Sun et al., 2024), this often incurs performance loss or GPU architecture-specific dependencies. Recent work on structured pruning in LLMs has addressed neuron width reduction: Ma et al. (2023) propose a "channel pruning" technique that removes neurons sharing the same index across all LLM layers, effectively reducing the width of the entire model. Ashkboos et al. (2024a) use PCA re-projections of neurons to prepare to remove low-eigenvalue associated neurons in the re-projected eigenspace. Gao et al. (2024b) reduce neuron width in a dimension-independent manner by training a separate GRU hypernetwork to predict neurons to remove. We depart from these pruning approaches by instead focusing on merging-based strategies.

Neuron-level redundancy Current LLMs often redundantly encode concepts in their representations, which has been repeatedly identified in prior work. Early work addressed this by wiring similar neurons together based on their weight set similarities (Srinivas & Babu, 2015) or using cluster-based pruning to remove redundant structures (Dalvi et al., 2020). In a model merging method, Stoica et al. (2024) further leveraged this by aligning and merging similar neurons both across and *within* models. Nanda et al. (2023) characterize redundancy in pre-trained models as "diffuse," where multiple random subsets of neurons in a model layer can approximate the performance of the whole layer. In our work, we are inspired by this diffusely redundant nature of neurons, and design our technique to exploit this redundancy. Theus et al. (2024) use optimal transport to merge groups of neurons specifically in ResNets, where only certain residual groups undergo optimal-transport based compression. We extend ideas in this approach to fully residually-connected LLMs, and operate at a layer level rather than group-level.

B. Simple explanation of QR Decomposition invariance

From Section 2, we provide an equality statement of RMSNorm applied to invertible maps. We include a short and simple proof for completeness. As a reminder, for model dimension d_{o} we have input vector $\mathbf{x} \in \mathbb{R}^{d_{\text{o}}}$, Q denotes orthogonal matrices where $Q^TQ = QQ^T = I$, $T = QR$ via QR decomposition, and T is invertible.

$$\begin{aligned}
\text{RMSNorm}(\mathbf{x}Q)RT^{-1} &= \frac{\mathbf{x}Q}{\|\mathbf{x}Q\|}RT^{-1} && \text{via RMSNorm definition} && (1)\\
&= \frac{\mathbf{x}QRT^{-1}}{\|\mathbf{x}\|} && \text{via } \|xQ\| = \sqrt{xQ^TQx} = \|x\| && (2)\\
&= \text{RMSNorm}(\mathbf{x}) && \text{via } QR = T \text{ substitution} && (3)
\end{aligned}$$

Given this result, we see that the RMSNorm is invariant to our constructed maps up to orthogonal projection onto the column space of Q. While in the case of square T this equality holds, our transport maps are rectangular due to the intentional difference in the width of source and target supports. Rectangular T is not invertible, meaning that we instead use the pseudoinverse instead in our method. We show that up to orthogonal projection onto $\text{col}(Q)$, this equality holds in Appendix C, and in practice, this modification is vital for stability; without it, our method fails to converge as the preservation of layer normalization is lost. Despite our specific application of these maps in our setting, we believe this QR modification to be broadly useful in other methods that transform models in-place using invariant or approximately invariant maps like rotation-based quantization and model merging (Liu et al., 2025; Imfeld et al., 2024).

C. QR Decomposition invariance for rectangular maps

We will prove a theorem related to the RMSNorm, and then discuss what the result dictates in terms of invariance to RMSNorm in the rectangular transport map T case.

Theorem C.1 (QR invariance for rectangular maps)**.** *Let $T \in \mathbb{R}^{d_o \times d_n}$ be QR-decomposed into $T = QR$, where semi-orthogonal $Q \in \mathbb{R}^{d_o \times d_n}$ has orthonormal columns ($Q^TQ = I_{d_n}$) and $R \in \mathbb{R}^{d_n \times d_n}$ is upper triangular and invertible. Then,*

$$\textit{RMSNorm}(xQ)RT^{\dagger} = \frac{xQQ^T}{\|xQQ^T\|}. \tag{4}$$

Proof. We first note that Q is semi-orthogonal, therefore

$$\|xQ\| = \sqrt{xQQ^Tx^T} = \sqrt{xQ(Q^TQ)Q^Tx^T} = \|xQQ^T\|. \tag{5}$$

Next, using the QR decomposition of T, the Moore–Penrose pseudoinverse is

$$\begin{aligned} T^\dagger &= (T^TT)^{-1}T^T \\ &= (R^TQ^TQR)^{-1}R^TQ^T \\ &= (R^TR)^{-1}R^TQ^T \\ &= R^{-1}Q^T. \end{aligned}$$

Substituting into the expression of interest gives

$$\text{RMSNorm}(xQ)RT^\dagger = \frac{xQ}{\|xQ\|}R(R^{-1}Q^T) \tag{6}$$

$$= \frac{xQQ^T}{\|xQ\|} \tag{7}$$

$$= \frac{xQQ^T}{\|xQQ^T\|}, \tag{8}$$

which completes the proof. □

D. Application of Maps

With the QR modification of our transportation maps for compression, we are able to apply our transformations at two key locations for each Transformer layer (Vaswani et al., 2017). We label our transformations per layer as $\{M_A, M_A^{\text{inv}}\}$ and $\{M_F, M_F^{\text{inv}}\}$.

We apply transformations to the standard linear projections in attention (W_Q, W_K, W_V, W_O) and feed-forward layers ($W_{\text{up}}, W_{\text{gate}}, W_{\text{down}}$) as defined in modern LLMs. We denote σ as the nonlinearity in GLU-based activations and $\odot$ for the gating operation (Shazeer, 2020). We label each of the linear weight matrices in modern LLMs as the following:

Per layer, we apply $\{M_A, M_A^{\text{inv}}\}$ and $\{M_F, M_F^{\text{inv}}\}$ to the attention and FFN blocks, respectively. Attention output is transformed by M_A, followed by M_A^{inv} before the FFN's application. Similarly, M_F transforms the FFN output, with M_F^{inv} preceding the next attention block. Assuming input $x_{\text{in}}^{\text{attn}}$ is pre-multiplied by M_F (from a prior layer) to give $M_F x_{\text{in}}^{\text{attn}} = x_{\text{in}}^{\text{attn}\prime}$, we have:

$$\begin{aligned} x_{\text{out}}^{\text{attn}} = W_O\text{MHA}(&W_Q M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}, \\ &W_K M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}, \\ &W_V M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}) + M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime} \end{aligned} \tag{9}$$

This reverses the effect of pre-multiplying $x_{\text{in}}^{\text{attn}}$ by M_{F} from the prior layer. To apply the M_A compression map, we apply it to $x_{\text{out}}^{\text{attn}}$, resulting in:

$$M_A x_{\text{out}}^{\text{attn}} = M_A W_O\text{MHA}(W_Q M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}, W_K M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}, W_V M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime}) + M_A M_F^{\text{inv}} x_{\text{in}}^{\text{attn}\prime} \tag{10}$$

Now, to reverse the effect of M_A on the attention output, which we will denote as $x_{\text{in}}^{\text{ff}\prime} = M_A x_{\text{out}}^{\text{attn}}$, we multiply our feed-forward input by M_A^{inv}:

$$x_{\text{out}}^{\text{ff}} = W_{\text{down}}\sigma(W_{\text{up}} M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime} \odot W_{\text{gate}} M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime}) + M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime} \tag{11}$$

And finally, to apply the M_F compression map, we have

$$M_F x_{\text{out}}^{\text{ff}} = M_F W_{\text{down}}\sigma(W_{\text{up}} M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime} \odot W_{\text{gate}} M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime}) + M_F M_A^{\text{inv}} x_{\text{in}}^{\text{ff}\prime} \tag{12}$$

All M matrices can be "absorbed" into neighboring weight matrices W due to the associativity of matrix multiplication. This application of maps mirrors several prior methods using a similar project/unproject scheme for Transformer model merging and compression (Imfeld et al., 2024; Ashkboos et al., 2024a; Verma & Elbayad, 2024).

E. Full zero-shot performance

Full zero-shot performance across pruning methods and +DOTRESIZE in addition to averages are found in Table 3.

Model	Sparsity	Method	ARC-C	ARC-E	HellaSwag	PIQA	Winogrande	Average
Llama-3.1-8B	0%	-	53.50	81.14	78.88	81.23	73.48	73.65
	20%	SliceGPT	40.61	68.22	59.53	73.23	62.12	60.74
		+DOTRESIZE	39.42	67.63	60.13	73.01	64.48	**60.94**
	30%	SliceGPT	34.13	58.67	45.73	67.03	53.91	51.89
		+DOTRESIZE	34.47	57.41	47.53	67.90	55.96	**52.65**
Llama-3.1-70B	0%	-	65.10	86.66	84.94	84.28	79.56	80.11
	20%	SliceGPT	55.20	80.43	69.19	79.65	72.93	**71.48**
		+DOTRESIZE	45.39	69.28	69.28	74.59	72.22	67.43
	30%	SliceGPT	50.26	73.91	63.85	77.20	70.24	**67.09**
		+DOTRESIZE	49.91	73.61	56.19	76.28	68.98	65.00
Mistral-7B	0%	-	52.22	78.20	80.47	82.26	73.80	73.39
	20%	SliceGPT	42.83	69.91	61.80	75.57	64.33	62.89
		+DOTRESIZE	42.66	69.95	62.31	75.35	65.43	**63.14**
	30%	SliceGPT	36.18	60.69	49.04	70.02	58.88	**54.96**
		+DOTRESIZE	34.04	58.54	47.65	69.10	59.04	53.67
Mistral-12B	0%	-	57.85	81.65	82.79	82.37	73.64	75.66
	20%	SliceGPT	29.86	46.51	53.19	68.17	59.43	51.43
		+DOTRESIZE	43.94	71.09	60.88	75.24	64.56	**63.14**
	30%	SliceGPT	26.96	40.24	42.88	63.11	56.51	45.94
		+DOTRESIZE	33.96	57.83	46.71	69.26	60.77	**53.71**
Phi-4 14B	0%	-	56.06	72.77	81.90	81.34	76.72	73.76
	20%	SliceGPT	54.61	76.09	73.08	80.30	71.11	71.04
		+DOTRESIZE	56.23	79.29	72.65	79.76	72.45	**72.08**
	30%	SliceGPT	49.83	74.12	64.57	76.33	69.38	**66.84**
		+DOTRESIZE	48.55	75.17	63.39	76.44	68.67	66.44

Table 3. Zero shot performance (% Accuracy) of models on downstream tasks under different width removal settings. Bolded results reflect the higher result on model and sparsity comparisons.

F. Layer-wise analysis of effective rank

For a data matrix of n items of dimension d_{orig}, we compute effective rank as the following, assuming we have ordered eigenvalues $\{\sigma_i\}_{i\in[1,d_{\text{orig}}]}$, and threshold $0 < \tau < 1$:

$$k = \arg\min_{k'} \frac{\sum_{i=1}^{k'} \sigma_i}{\sum_{i=1}^{d_{\text{orig}}} \sigma_i} > \tau$$

In other words, we find the minimum index k such that the cumulative sum of the first k eigenvalues exceeds ratio τ. In this analysis, we fix $\tau = 0.99$. We compute the effective rank for two models, Llama-3.1-8B and Mistral-7B across all 32 layers, at both the output of the attention sublayer and the output of the MLP sublayer. Both models have true model dimension of

4096. We report our results in Figure 3, and find that Mistral has significantly reduced effective rank for many early layers as compared to Llama, suggesting that Llama has a more dispersed signal across its PCA-reprojected neurons.

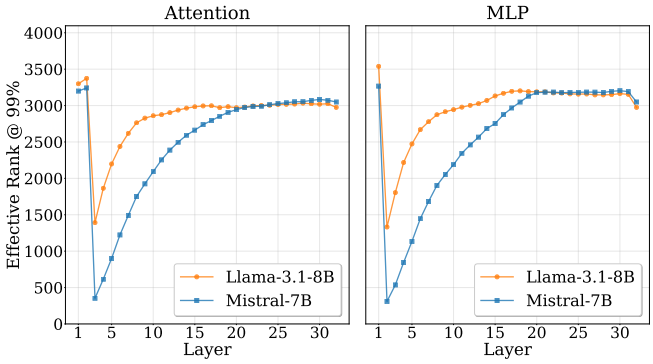

Figure 3. Effective rank at 99% variance of Llama-3.1-8B and Mistral-7B activations at each layer during PCA+DOTRESIZE application for 20% width reduction. We report values for both the output of attention and the MLP, after the residual connection. We find that Mistral reaches a much smaller effective rank in earlier layers, suggesting that it contains less distributed signal across neurons after PCA reprojection.

G. Analysis

Compute cost versus performance As our method is intended to reduce memory use and improve latency, we measure the total compute used in inference: we find the minimum number of 32GB V100 GPUs the compressed Llama-3.1-70B model can run on without GPU-offloading, and measure the latency in ms/token during inference on sequences of 128 tokens (Ashkboos et al., 2024a). This gives us the total compute cost in GPUms/token. We compress at different levels of width reduction, and evaluate both perplexity and inference cost, and report this trade-off in Figure 4. After 20% width reduction, the overall latency begins to improve; the small region of increased cost is incidental to the cost involved in reprojecting the residual stream in SliceGPT and correspondingly our method.

Sinkhorn regularization parameter We select $\lambda = 0.1$ as the regularization parameter for Sinkhorn-based OT. To understand the effect of different values of λ, we evaluate three of our models on 5 different λ values, ranging from 0.01 to 10. As seen in Figure 5, the amount of regularization does not significantly impact downstream language modeling performance, suggesting that multiple weighting strategies can successfully recombine neurons. However, minor variations in λ can result in some perplexity differences in the Llama-3.1-8B model as seen at $\lambda = 0.01$. While optimal λ values may vary due to training differences across models, the results presented here indicate a broad range of effective settings.

Calibration data Similar to prior work that examines the behavior of neurons for alignment or pruning (Singh & Jaggi, 2020; Ainsworth et al., 2023; Ashkboos et al., 2024a), our method uses a small amount of relevant calibration data (128 samples) to determine the behavior of individual neurons. We evaluate the sensitivity of our method to the amount of calibration data used by testing the use of 32, 64, 128, and 256 samples to compute transportation maps. Results are shown in Figure 6. We find that while our method can improve with additional calibration data, it is fairly robust to the number of samples used. This robustness is practical, as computing the cost matrices C for optimal transport can become computationally expensive with large datasets, and suggests that a small amount of exemplar data is sufficient to capture meaningful neuron behavior.

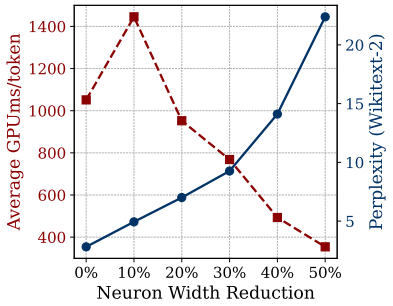

Figure 4. Width reduction vs performance tradeoff across different levels of compression on Llama-3.1-70B. After 20%, real world compute cost reduction is observed.

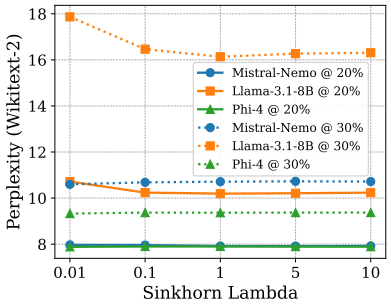

Figure 5. Performance on Wikitext-2 at 20% and 30% width reduction with different entropic regularization λ. Our method is not sensitive to the amount of regularization for these models.

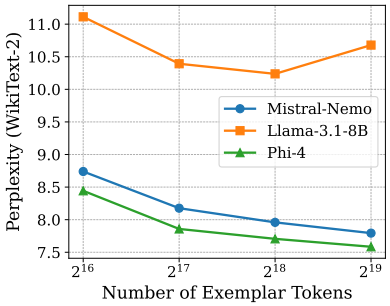

Figure 6. Performance at 20% sparsity with varying exemplar tokens. After approximately 2^{17} tokens, the returns on using more calibration data appear diminishing.