Model Merging by Output-Space Projection

Bethan Evans [1] **Benjamin Etheridge** [2] **Stephen Roberts** [2] **Jared Tanner** [1]

Abstract

Model merging combines fine-tuned checkpoints into a single multi-task model without retraining. Existing methods—such as task arithmetic, model soups, TIES, and DARE—are computationally efficient and empirically successful, but rely on heuristic design choices and lack formal optimality guarantees. We show that merging can be formulated as a convex quadratic programme over residual updates, yielding weights that minimise a squared-output calibration objective using calibration inputs and fine-tuned model outputs, and subsuming existing methods as special cases. Our framework yields a closed-form diagnostic—the fraction of residual energy captured by a chosen basis—that predicts downstream merge quality using only the calibration set. Empirically, the QP matches or outperforms existing methods in the single-layer setting, and we characterise when the optimal basis provides significant gains over the cheaper diagonal QP. We extend to multi-layer merging via a sequential layer-wise algorithm and demonstrate consistent gains across language and vision benchmarks.

1. Introduction

The proliferation of task-specific fine-tuned language and vision models has motivated growing interest in model merging to improve performance across broader sets of tasks; see (Yang et al., 2025) and references therein. Existing practical approaches—task arithmetic (Ilharco et al., 2023a), model soups (Wortsman et al., 2022a), TIES (Yadav et al., 2023), DARE (Yu et al., 2023), Fisher (Matena and Raffel, 2022a) and Adamerge (Yang et al., 2024)—combine residual updates using fixed or stochastic rules and have demonstrated strong empirical performance. However, these methods do not characterise the optimal combination weights, nor have a precise mathematical notion or loss function for which their method is optimal.

We begin with a simple observation: model merging via diagonal masks on fine-tuned residuals can be cast as a convex quadratic programme (QP). Solving this QP yields weights that minimise a squared-output calibration objective using calibration inputs and fine-tuned model outputs, and subsumes existing diagonal-mask methods—including task arithmetic and TIES—as special cases. This observation can be extended to a more general principle. Rather than combining parameters directly, merging can be viewed as approximating the residual errors of a base model using a subspace of candidate corrections. This leads to a formulation in which merging corresponds to projecting residuals onto an admissible subspace in output space. The corresponding minimum-loss solution is obtained by projecting base-model residuals onto the subspace spanned by candidate model updates, where the optimal basis for the relaxed projection problem is given by the leading eigenvectors of the residual energy matrix.

This projection perspective provides a unifying framework for a broad class of quadratic and basis-dependent model merging methods. By generalising the diagonal QP to arbitrary orthonormal bases, we obtain a general basis QP and derive a closed-form expression for the irreducible error of any basis choice. Minimising this error yields the output-weighted principal components of residual activations on the calibration set as the optimal basis, with the diagonal basis arising as a special case. While the diagonal QP is computationally efficient, the optimal basis requires a data-dependent eigendecomposition of the residual matrix, yielding lower squared-output loss at higher cost. Our framework characterises this trade-off through a computable suboptimality gap. In particular, diagonal merging is optimal when model updates decompose into independent modes in a shared basis. Our framework also clarifies the relationship to parameter-space methods. We show that existing approaches such as model soups, TIES, and task arithmetic correspond to special cases of a diagonal-mask QP in a chosen basis, and we demonstrate that our diagonal QP is optimal on some popular language and image models.

[1]Department of Mathematics, University of Oxford, Oxford, UK [2]Department of Engineering Science, University of Oxford, Oxford, UK. Correspondence to: Bethan Evans <bethan.evans@maths.ox.ac.uk>.

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).

2. Background

We consider an M-layer linear network with weight matrices $\{W_i\}_{i=1}^{M}$. For input $x \in \mathbb{R}^d$ the base-model output is

$$h(x;\theta) \;=\; W_M W_{M-1} \cdots W_1\, x \in \mathbb{R}^c. \tag{1}$$

Suppose we are given K fine-tuned models obtained by adding a weight residual update to a single layer N. For model $k \in \{1,\dots,K\}$ the update is $\delta_N^{(k)}$, yielding

$$h(x;\theta+\delta_N^{(k)}) \;=\; W_M \cdots (W_N+\delta_N^{(k)}) \cdots W_1\, x. \tag{2}$$

Initially, we focus on single-layer merging at layer N; multi-layer extensions are discussed later. For notational convenience, define the lower and upper compositions

$$Z \;:=\; \prod_{\ell=1}^{N-1} W_\ell, \qquad L \;:=\; \prod_{\ell=N+1}^{M} W_\ell, \tag{3}$$

so that the model factors as

$$h(x;\theta) = LW_NZx, \qquad h^{(k)}(x) = L(W_N+\delta_N^{(k)})Zx.$$

Given a calibration set $\{(x_j,y_j)\}_{j\in\mathcal{K}}$, where y_j denotes the desired output for input x_j, our goal is to combine the residual updates $\{\delta_N^{(k)}\}_{k=1}^{K}$ into a single merged model that performs well across tasks. In practice, the method requires only calibration inputs together with outputs from the fine-tuned models, so y_j may instead be taken from the corresponding task-specific model when labelled calibration data is unavailable. Throughout, the squared output loss is chosen because it forms the loss as a convex quadratic. Under the same linear-in-weight-update assumption, cross-entropy yields a convex but non-quadratic objective in the merge coefficients. It therefore does not admit the same closed-form projection and trace-based characterisation; we leave a systematic treatment of non-quadratic losses to future work.

Extension to Non-Linear Networks The linear assumption may be relaxed by linearising the merged layer of the network around the base model, so that the output is approximately linear in the residual update. This approximation is exact for linear models and locally accurate for sufficiently small updates in some piecewise-linear networks.

3. Diagonal Mask Merging as a QP

We propose to merge models by applying a *diagonal mask* $D_k = \mathrm{diag}(d^{(k)}) \in \mathbb{R}^{r\times r}$, $d^{(k)} \in \mathbb{R}^r$, to each residual before summing. Writing the stacked decision vector $\mathbf{d} = [d^{(1)};\dots;d^{(K)}] \in \mathbb{R}^{Kr}$, the merged residual at layer N is

$$\delta_{\mathrm{merge}}(\mathbf{d}) \;:=\; \sum_{k=1}^{K} D_k\, \delta_N^{(k)}, \tag{4}$$

and the corresponding merged-model output on input x is

$$h_{\mathrm{merge}}(x;\mathbf{d}) \;:=\; L(W_N+\delta_{\mathrm{merge}}(\mathbf{d}))Zx. \tag{5}$$

Existing merging methods fall under this framework as special cases: setting $D_k = \lambda_k I$ for fixed scalars λ_k recovers *task arithmetic* (Ilharco et al., 2023b); $D_k = (1/K)\,I$ recovers model soups (Wortsman et al., 2022b); $D_k = \mathrm{diag}(m^{(k)}/p)$ with $m_i^{(k)} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$ recovers a *row-uniform* restriction of DARE (Yu et al., 2023), in which all entries within a row of the residual are scaled by the same factor (the full element-wise DARE explores a strictly larger mask family); a diagonal mask with sign-based pruning can be interpreted as TIES (Yadav et al., 2023).

A detailed description of how common existing methods fit into this framework is given in App. A.

For a calibration set $\mathcal{K}$ define the squared-output loss by:

$$\mathcal{J}(\mathbf{d}) \;:=\; \sum_{j\in\mathcal{K}} \left\| A^{(j)}\mathbf{d} + b_j \right\|^2, \tag{6}$$

where $b_j := h(x_j;\theta) - y_j$ and $A^{(j)}$ is formed as follows: the hidden-layer residual $r_{i,j} := \delta_N^{(i)} Z x_j \in \mathbb{R}^r$ is diagonalised to form $A_{i,j} := L\,\mathrm{diag}(r_{i,j})$, and these are concatenated as $A^{(j)} := [A^{(1,j)} \;\cdots\; A^{(K,j)}] \in \mathbb{R}^{c\times Kr}$. Here y_j denotes either labelled calibration targets, when available, or outputs from the corresponding fine-tuned model. Expanding gives the convex quadratic

$$\mathcal{J}(\mathbf{d}) \;=\; \tfrac{1}{2}\mathbf{d}^\top H\mathbf{d} + g^\top\mathbf{d} + \mathrm{const}, \tag{7}$$

where we define

$$H \;=\; 2\sum_{j\in\mathcal{K}} (A^{(j)})^\top A^{(j)} \;\succeq\; 0, \tag{8}$$

$$g \;=\; 2\sum_{j\in\mathcal{K}} (A^{(j)})^\top b_j. \tag{9}$$

Subject to any convex constraint set $\mathcal{C}$ (e.g. box constraints $0 \le \mathbf{d} \le \mathbf{1}$), the resulting programme

$$\min_{\mathbf{d}\in\mathcal{C}} \;\tfrac{1}{2}\mathbf{d}^\top H\mathbf{d} + g^\top\mathbf{d} \tag{QP}$$

is convex and can be solved globally (Boyd and Vandenberghe, 2004) to obtain the optimal diagonal mask vector $\mathbf{d}^*$. Since methods such as task arithmetic and model soups correspond to feasible points of (QP), the optimal solution achieves validation loss no worse than these methods under the same objective.

4. General Basis Irreducible Error

The QP formed by this merging strategy is convex and thus attains a global minimum; however, the achievable loss depends on the chosen subspace. We extend this to a general

basis and show that there exists a basis minimising the relaxed irreducible error in the diagonal-mask framework. The diagonal merging introduced in Section 3 corresponds to restricting updates to the standard coordinate basis $\{e_i\}_{i=1}^r$ in the residual space. Left-multiplication by a diagonal mask scales the rows of the residual update independently, corresponding to weighting hidden features at the merged layer without mixing coordinates. The resulting output corrections therefore lie in the subspace $\mathrm{span}\{Le_i\}$.

We now consider a relaxed projection problem in which each residual may be corrected by an arbitrary vector in a common output subspace $\mathcal{S} \subseteq \mathbb{R}^c$. This yields a closed-form characterisation of how much residual energy a chosen output subspace can capture. In the relaxed formulation, each sample residual may be corrected independently within the subspace. The practical merging problem instead requires a shared set of merge coefficients across all samples, leading to the general-basis QP in Proposition E.1. We consider any orthonormal basis $P = \{q_1, \dots q_p\}$ of the subspace.

Theorem 4.1 (Merging as a Projection and Basis-Dependent Error)**.** *Let $\mathcal{K}$ be the calibration set with $n := |\mathcal{K}|$, and let $h(x_j;\theta)$ denote the base model with residuals $b_j = h(x_j;\theta) - y_j \in \mathbb{R}^c$. Define the matrix*

$$S := \sum_{j\in\mathcal{K}} b_j b_j^\top \in \mathbb{R}^{c\times c}.$$

Let the set of all output corrections the parametrisation can produce be denoted by $\mathcal{S} \subseteq \mathbb{R}^c$ for the whole calibration set (the range of the linear map from coefficients to stacked outputs). Let $P_\mathcal{S}$ be the orthogonal projector onto $\mathcal{S}$. The optimal correction inside $\mathcal{S}$ is the orthogonal projection of $-b_j$ onto $\mathcal{S}$, so the residual after merging for sample j is

$$(I - P_\mathcal{S})b_j.$$

Recall that $\mathcal{J}(\mathbf{d})$ denotes the squared-output loss as a function of the merge coefficients $\mathbf{d}$, and let $\mathcal{J}_\mathcal{S}$ correspond to the minimum of $\mathcal{J}(\mathbf{d})$ over all $\mathbf{d}$ whose induced output lies in $\mathcal{S}$. Using the identity $\sum_{j\in\mathcal{K}} \|P_\mathcal{S} b_j\|^2 = \mathrm{tr}(SP_\mathcal{S})$, we obtain that the minimal loss on the calibration set $\mathcal{K}$ attainable within $\mathcal{S}$ is

$$\mathcal{J}_\mathcal{S} \;=\; \sum_{j\in\mathcal{K}} \|b_j\|^2 \;-\; \mathrm{tr}(S\,P_\mathcal{S}). \tag{10}$$

The proof of Theorem 4.1 is provided in Appendix B.1. Since $\mathrm{tr}(S) = \sum_{j\in\mathcal{K}} \|b_j\|^2$ is the total residual energy, we can normalise (10) to obtain

$$\frac{\mathcal{J}_\mathcal{S}}{\mathrm{tr}(S)} = 1 - \frac{\mathrm{tr}(SP_\mathcal{S})}{\mathrm{tr}(S)}. \tag{11}$$

Thus, $\mathrm{tr}(SP_\mathcal{S})/\,\mathrm{tr}(S)$ is a data-dependent diagnostic of how well an output subspace captures residual energy. Since $\sum_j \|b_j\|^2$ is independent of the choice of $\mathcal{S}$, selecting the best output subspace of fixed dimension reduces to maximising the captured energy $\sum_j \|P_\mathcal{S} b_j\|^2$. This yields a basis-dependent characterisation of the error. We show empirically in Section 5 that this fraction predicts downstream performance for an image classifier.

Corollary 4.2. *Minimising the loss over admissible p-dimensional subspaces $\mathcal{S} \subseteq \mathbb{R}^c$ is equivalent to maximising $\mathrm{tr}(SP_\mathcal{S})$. Among all p-dimensional subspaces of $\mathbb{R}^c$, the minimum-loss subspace for the relaxed projection problem is given by the span of the top-p eigenvectors of matrix S.*

Proof. Follows from the Ky-Fan variational characterisation of trace maxima of symmetric PSD matrices over rank-p projectors (Eckart and Young, 1936). □

The minimum-loss subspace identified by Cor. 4.2 is defined over all subspaces of $\mathbb{R}^c$. In practice, however, the model parametrisation restricts admissible corrections to a subset of this space that can be realised in the chosen basis. In our setting, a chosen basis $\{q_i\}_{i=1}^p$ induces the output subspace

$$\mathcal{S}_{\mathrm{model}} := \mathrm{span}\{Lq_1, \dots, Lq_p\}.$$

This subspace describes the output directions available to corrections expressed in the chosen basis. The projection analysis above therefore provides a basis-dependent measure of how much residual energy these directions can capture. In general, however, $\mathcal{S}_{\mathrm{model}}$ may fail to contain the principal residual directions identified by the projection analysis. The component of the residual energy lying outside this pointwise output subspace therefore gives a basis-dependent irreducible error for the relaxed projection problem. Diagonal masking is optimal for the relaxed projection problem when the output subspace $\mathrm{span}\{Le_1, \dots, Le_p\}$ coincides with the top-p eigenspace of S.

In the special case $L = I$, this reduces to selecting the p coordinates with largest diagonal entries when S is diagonal in the standard basis. The excess squared-output loss on calibration set $\mathcal{K}$ incurred by merging using a p-dimensional subspace $\mathcal{S}_{\mathrm{model}}$ instead of the optimal subspace $\mathcal{S}_{\mathrm{opt}}$ is

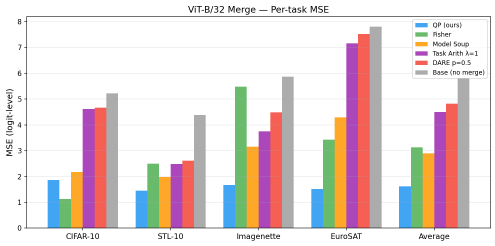
$$\mathcal{J}_{\mathcal{S}_{\mathrm{model}}} - \mathcal{J}_{\mathcal{S}_{\mathrm{opt}}} = \mathrm{tr}\left(S(P_{\mathrm{opt}} - P_{\mathrm{model}})\right). \tag{12}$$

This quantity is computable from the calibration set and provides a data-dependent measure of the suboptimality of the chosen basis. The residuals define an anisotropic energy distribution in output space whose principal directions are given by the eigenvectors of the residual matrix S. Merging therefore corresponds to selecting a subspace that captures as much residual energy as possible. Diagonal masking is efficient but becomes suboptimal when this energy is not axis-aligned; Thm. 4.1 quantifies the resulting loss. A corresponding optimisation in a general basis, allowing the basis to minimise this irreducible error, is given in App. E.

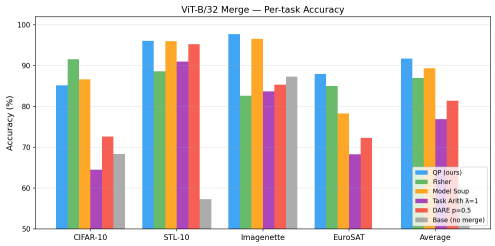
5. Experiments

We compare our theoretical results to empirical benchmarks and demonstrate the use of irreducible error in the chosen basis to predict merging success. A detailed experimental setup is given in App. D.

We begin with a CLIP-pretrained ViT-B/32 model and fine-tune only the final transformer-block MLP output projection on four downstream datasets: CIFAR-10, STL-10, Imagenette, and EuroSAT. We then merge the resulting task-specific models and evaluate MSE and accuracy on each dataset and on the combined benchmark. The QP is solved approximately using 100 gradient steps. To construct the objective, we linearise the task-specific logit map around 100 calibration examples per task using the Jacobian of the downstream normalisation and head with respect to the merge-layer representation. Figure 1 shows that the diagonal QP achieves the lowest MSE across all datasets and the best average accuracy among the methods considered. The cross-entropy results in Figure 3 show similar behaviour. Additional single-layer vision experiments on MNIST with a 3-layer MLP are given in App. D.7, an extension to multi-layer merging via sequential application of the QP is given in App. D.6 and experiments on the LLM Llama 3.1 are given in App. D.9.



(a) MSE on ViT-B/32.



(b) Accuracy on ViT-B/32.

Figure 1. Performance on ViT-32 across single-layer merging methods.

Geometry Experiment Recall that the residuals on a calibration set define the matrix $S = \sum_j b_j b_j^\top$, where $b_j = h_0(x_j) - y_j$ is the difference between the base model output and the target output. Our theory predicts that, in the relaxed output-subspace problem, the optimal p-dimensional subspace is spanned by the top-p eigenvectors of S. We evaluate merging under several choices of basis: the optimal basis (top-p eigenvectors of S), the standard basis (diagonal masking), the SVD basis of the residual updates, and random orthonormal bases.

Figure 2 plots the fraction of captured residual energy, $\mathrm{tr}(SP_{\mathcal{S}})/\mathrm{tr}(S)$, against MSE as the basis dimension p increases. As p increases, the captured residual energy increases and the MSE decreases, consistent with the projection analysis. Figure 2 also compares different bases on five fine-tuned 3-layer ReLU networks trained on separate MNIST digit classification tasks, merging at the final layer. For each basis, we restrict merging to the same dimension p and solve the corresponding QP. The optimal basis consistently achieves the lowest MSE, while random bases perform substantially worse. The standard diagonal basis is less effective than the optimal basis, but remains competitive and outperforms the benchmark merging heuristics tested, indicating that the diagonal QP provides an efficient proxy when computing the optimal basis is impractical.

5.1. Conclusion

Our framework is derived for a linearised squared-output objective and a single merged layer; a sequential multi-layer extension is given in App. F. The optimal-basis QP additionally requires an eigendecomposition of the residual matrix S, while the cheaper diagonal QP may be suboptimal when residual energy is not axis-aligned. Moreover, the energy-fraction diagnostic is derived from a relaxed projection problem and therefore only approximates attainable downstream loss. Our experiments primarily illustrate how the framework helps explain when existing merging heuristics succeed, when diagonal merging is sufficient, and when basis misalignment leads to degradation. Extending the framework to non-quadratic losses and developing scalable approximations to the optimal basis remain important directions for future work.

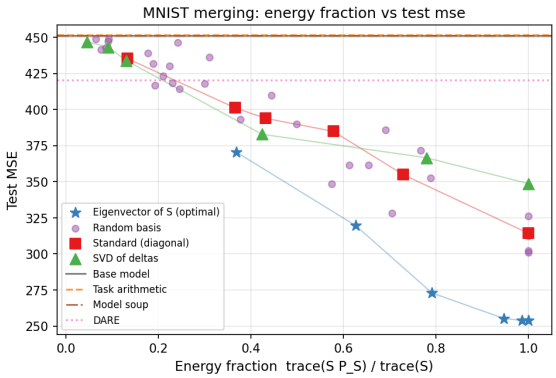


Figure 2. MSE against captured energy for increasing basis dimensions p on MNIST.

Acknowledgements

The authors acknowledge support from His Majesty's Government in the development of this research. Jared Tanner is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through the grant EP/Y028872/1. The authors are also grateful to Josh Collyer of the Alan Turing Institute, whose presentation on quantization results at the AI Security Reading Group inspired this work.

References

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 978-0-521-83378-3. Google-Books-ID: mYm0bLd3fcoC.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, July 2021. URL http://arxiv.org/abs/2107.03374. arXiv:2107.03374 [cs].

Adam Coates, Honglak Lee, and Andrew Y Ng. An Analysis of Single-Layer Networks in Unsupervised Feature Learning.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs].

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/document/5206848. ISSN: 1063-6919.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators, April 2024. URL https://arxiv.org/abs/2404.04475v2.

Carl Eckart and Gale Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3): 211–218, September 1936. ISSN 0033-3123, 1860-0980. doi: 10.1007/BF02288367.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, August 2017. doi: 10.1109/JSTARS.2019.2918242.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023a.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing Models with Task Arithmetic, March 2023b. URL http://arxiv.org/abs/2212.04089. arXiv:2212.04089 [cs].

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengfei Chaudhury. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2023.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images.

Michael Matena and Colin Raffel. Merging Models with Fisher-Weighted Averaging, August 2022a. URL http://arxiv.org/abs/2111.09832. arXiv:2111.09832 [cs].

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. ZipIt! Merging Models from Different Tasks without Training, March 2024. URL http://arxiv.org/abs/2305.03053. arXiv:2305.03053 [cs].

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carlin, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, July 2022b. URL http://arxiv.org/abs/2203.05482. arXiv:2203.05482 [cs].

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive Model Merging for Multi-Task Learning, May 2024. URL http://arxiv.org/abs/2310.02575. arXiv:2310.02575 [cs].

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model Merging in LLMs, MLLMs, and Beyond: Methods, Theories, Applications and Opportunities, December 2025. URL http://arxiv.org/abs/2408.07666. arXiv:2408.07666 [cs].

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super Mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*, 2023.

A. Related Work

Task arithmetic and model soups. Ilharco et al. (2023a) introduced task vectors and showed that arithmetic operations on them transfer to model behaviour. Wortsman et al. (2022a) showed that averaging fine-tuned weights improves robustness. Both are recovered as special cases of our framework.

TIES and DARE. Yadav et al. (2023) introduced sign-based conflict resolution; Yu et al. (2023) introduced random dropping of residual parameters. We show that DARE with row-uniform masks is a special case of our framework.

Fisher-weighted merging. Matena and Raffel (2022b) weight parameters by their Fisher information. Our method is conceptually related to precision-weighted merging (Fisher), but operates in the singular basis of residual updates rather than parameter curvature. Fisher merging is a related parameter-space quadratic objective using Fisher precision, rather than an exact instance of our squared-output QP.

RegMean. RegMean (Jin et al., 2023) computes merge weights by solving a closed-form linear regression at each layer, using second-moment statistics of activations to match the layer-wise outputs of the fine-tuned models. It is a related layer-wise least-squares approach, but operates on parameter-space activation statistics rather than the output-space residual objective optimised here, and we do not claim it as a special case of our framework.

ZipIt!. ZipIt! (Stoica et al., 2024) merges models by identifying and aligning correlated features across networks, effectively constructing a shared representation before combining parameters. In our framework, this can be interpreted as a basis alignment step that transforms residual updates into a space where corresponding directions are more compatible. Unlike diagonal masking, which fixes the basis and optimises coefficients, ZipIt! modifies the representation itself to reduce misalignment between models. Thus, it can be viewed as complementary to our approach: ZipIt! seeks a favourable basis in which merging is easier, while our method optimises merging within a given basis.

Subspace and spectral methods. KnOTS aligns LoRA updates by stacking low-rank update matrices, performing an SVD to obtain a shared orthonormal basis, expressing each update in that basis, and merging coefficients in the aligned space. More broadly, this fits a family of low-rank and core-space alignment methods that choose a basis from the residual updates themselves; in our framework these correspond to choosing $\{q_p\}$ from the column space of stacked residuals rather than from the eigenbasis of S.

Method	Basis	Interpretation in QP
Task Arithmetic	Standard basis $\{e_i\}$	Fixed scalar weights; recovers uniform scaling in diagonal QP
Model Soup	Standard basis $\{e_i\}$	Uniform averaging; a fixed feasible point of the QP
DARE (row-restricted)	Standard basis $\{e_i\}$	Row-wise stochastic masking; structured sparsity, approximates diagonal QP
TIES	Standard basis $\{e_i\}$	Sign-based pruning; sparsity constraint, ignores coupling between coordinates
Diagonal QP (ours)	Standard basis $\{e_i\}$	Optimal weights under diagonal (axis-aligned) restriction
Optimal-basis QP (ours)	Eigenbasis of S	Optimal subspace under quadratic objective
SVD-based methods	Singular vectors of $\delta^{(k)}$	Basis alignment; partially decouples QP across directions
RegMean	Layer-wise activation statistics	Related layer-wise least-squares method; not an instance of the output-space QP
Fisher merging	Parameter-space Fisher metric	Related quadratic objective; not an exact squared-output QP special case
ZipIt!	Learned aligned basis	Feature alignment; modifies basis before merging

Table 1. Interpretation of common model merging methods within the proposed QP framework. Exact correspondences and approximations are distinguished in the text.

A.1. SVD-Basis

Now let us consider setting the basis to the minimal set of singular vectors of the fine-tuned residual weights in the QP. We consider the singular value decomposition (SVD) of each of the fine-tuned residuals, and use this to derive optimal scaling weights applied to each singular direction. We begin by optimising for one direction only, then extend to multiple directions.

For each model i let the SVD of the layer-N residual be

$$\delta_N^{(i)} \;=\; \sum_{\ell=1}^{r_i} \sigma_\ell^{(i)}\, u_\ell^{(i)}\, (v_\ell^{(i)})^\top, \tag{13}$$

where $r_i = \mathrm{rank}(\delta_N^{(i)})$. Fix a unit left singular direction $u = u_a^{(i)} \in \mathbb{R}^r$ from some model a.

Assumption A.1 (Shared direction)**.** Suppose there exists a set $T \subseteq [K]$ of models and an index ℓ such that every $i \in S$ shares u as its ℓ-th left singular vector with the same right singular vector v and for every $i \in [K] \setminus S$ all singular directions are orthogonal to u. We may write $\delta_N^{(i)} = \sigma^{(i)} u v^\top + R^{(i)}$ where $u^\top R^{(i)} = 0$ for all $i \in T$.

Assumption A.2 (Downstream isometry)**.** $L^\top L = I$, i.e. the downstream weight product is an isometry.

Under Assumption A.1, for $i \in T$:

$$m_i \;=\; \sigma^{(i)}\, (v)^\top Z x \;=:\; \sigma^{(i)}\, s, \tag{14}$$

where $s := (v)^\top Z x \in \mathbb{R}$ is a scalar common to all models.

Theorem A.3 (Signal-to-total-power merging weights)**.** *Under Assumptions A.1–A.2, if the target y_j was produced by model $j \in S$ then*

$$\beta \;=\; (Lu)^\top (h(x;\theta) - y_j) \;=\; -u^\top \delta_N^{(j)} Z x \;=\; -\sigma^{(j)}\, s. \tag{15}$$

and the 1-D optimal mask weight for model $k \in T$ when merging to match task j's output is

$$d_k^* \;=\; \frac{\sigma^{(j)}\, \sigma^{(k)}}{\sum_{\ell \in S} \big(\sigma^{(\ell)}\big)^2}. \tag{16}$$

When the optimal basis happens to align with the shared singular directions of the residuals, the per-direction QP decomposes exactly and Theorem A.3 gives the closed form solution for each block.

Corollary A.4 (Recovery of existing methods in the SVD Basis)**.** *Under the equal singular value case $\sigma^{(k)} = \sigma$ for all $k \in T$, $d_k^* = 1/|S|$, recovering model averaging. When $|T| = 1$ for every choice of u (i.e each model has unique and orthogonal singular vectors to eachother), then $d_k^* = 1$, recovering direct addition.*

B. Additional Proofs

B.1. Proof of Theorem 4.1

Proof. Let the set of all output corrections the parametrisation can produce be denoted by $\mathcal{S} \subseteq \mathbb{R}^c$ for the whole validation set (i.e. the range of the linear map from coefficients to stacked outputs) and $P_{\mathcal{S}}$ is the orthogonal projector onto $\mathcal{S}$. The optimal correction inside $\mathcal{S}$ is the orthogonal projection of $-b_j$ onto $\mathcal{S}$, hence the residual after merging for sample j is $(I - P_{\mathcal{S}})b_j$. Therefore the attained squared-output loss over validation set $\mathcal{K}$ is

$$\mathcal{J}_{\mathcal{S}} \;=\; \sum_{j \in \mathcal{K}} \big\|(I - P_{\mathcal{S}})b_j\big\|^2 \;=\; \sum_{j \in \mathcal{K}} \big(\|b_j\|^2 - \|P_{\mathcal{S}} b_j\|^2\big). \tag{17}$$

Since $\sum_j \|b_j\|^2$ is independent of the choice of $\mathcal{S}$, the problem of selecting the best output subspace $\mathcal{S}$ of fixed dimension reduces to maximising the captured energy $\sum_j \|P_{\mathcal{S}} b_j\|^2$. Using the cyclicity of trace we can rewrite the captured energy in terms of S and the projector:

$$\sum_{j \in \mathcal{K}} \|P_{\mathcal{S}} b_j\|^2 \;=\; \sum_{j \in \mathcal{K}} \mathrm{tr}\left(b_j b_j^\top P_{\mathcal{S}}\right) \;=\; \mathrm{tr}\left(S\, P_{\mathcal{S}}\right). \tag{18}$$

Hence we obtain that the minimal loss attainable on $\mathcal{S}$ is

$$\mathcal{J}_{\mathcal{S}} \;=\; \sum_{j\in\mathcal{K}} \|b_j\|^2 \;-\; \mathrm{tr}(S\,P_{\mathcal{S}}),$$

and the loss difference between two subspaces is:

$$\mathcal{J}_{P_2} - \mathcal{J}_{P_1} = \mathrm{tr}(S(P_1 - P_2)).$$

□

C. Single Direction General Basis Case

First we consider a single direction q. Define the *scalar projection* of model i's residual activation onto q for input x:

$$m_i \;:=\; q^\top \delta_N^{(i)} Z x \;\in\; \mathbb{R}, \tag{19}$$

and the projection of the base-model residual onto the downstream direction Lq:

$$\beta \;:=\; (Lq)^\top b, \qquad b \;:=\; h(x;\theta) - y. \tag{20}$$

Restricting the objective to the component of outputs along q, the 1-D objective for a single data point (x, y) is

$$\mathcal{J}(\mathbf{d}) \;=\; \Big(\sum_{i=1}^{K} d_i\, m_i + \beta\Big)^2 \;=\; \big(\mathbf{d}^\top \mathbf{m} + \beta\big)^2, \tag{21}$$

with $\mathbf{d} = (d_1, \ldots, d_K)^\top$ and $\mathbf{m} = (m_1, \ldots, m_K)^\top \in \mathbb{R}^K$.

Proposition C.1 (1-D optimal mask)**.** *If $\mathbf{m} \neq 0$, the minimum-norm minimiser of* (21) *over $\mathbf{d} \in \mathbb{R}^K$ is*

$$\mathbf{d}^* \;=\; -\frac{\beta\,\mathbf{m}}{\|\mathbf{m}\|^2}. \tag{22}$$

If $\mathbf{m} = 0$, the objective is constant and every $\mathbf{d}$ is optimal.

Thus we see that the diagonal mask value is given by an energy weighting of the fine-tuning in this direction on a given input as well as the back-propagated energy in the output from that direction.

D. Experimental Set-Up

Setup: We implement our merging experiments using Python 3.12.3 and PyTorch 2.7.0 that supports CUDA 12.8 for accelerating computations by using GPUs. We run our experiments on a machine equipped with an Intel Xeon Gold 5418Y 2.00GHz 24-core processor, 1.5TiB of RAM, and four NVIDIA H100 NVL GPUs.

D.1. Datasets

Here we give an overview of the datasets used.

CIFAR-10 CIFAR-10 is a balanced image classification benchmark comprising 60,000 colour images drawn from ten object categories (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck) (Krizhevsky). We use the standard split of 50,000 training images (5,000 per class) and 10,000 test images (1,000 per class). Each image is an RGB photograph of fixed size 32×32 pixels, giving 1,024 pixels per image and, with three colour channels, 3,072 channel values in total. Pixel intensities are stored as 8-bit unsigned integers in the range $[0, 255]$.

STL-10 STL-10 (Coates et al.) is an image recognition benchmark comprising 10 object classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). It contains colour images of size 96×96 pixels, with 5,000 labelled training images (500 per class), 8,000 test images (800 per class), and an additional 100,000 unlabelled images for unsupervised learning.

Imagenette Imagenette is a smaller subset of 10 easily classified ImageNet (Deng et al., 2009) classes designed for rapid experimentation. We use the standard v2 split, which contains 9,469 training images and 3,925 validation images in the full-size variant. The 160 px and 320 px variants resize the shortest side to the chosen resolution while preserving aspect ratio.

EuroSAT EuroSAT (Helber et al., 2017) is a land use and land cover classification benchmark based on Sentinel-2 satellite imagery. It consists of 27,000 labelled and geo-referenced image patches across 10 classes, with each patch measuring 64×64 pixels. The dataset is available in both RGB and multispectral forms; the multispectral version covers 13 spectral bands.

MMLU Massive Multitask Language Understanding (MMLU) is a holistic benchmark designed to measure world knowledge and problem-solving across 57 subjects spanning STEM, the humanities, the social sciences, and more. The tasks range in difficulty from elementary to advanced professional levels. We evaluate models using 5-shot accuracy on the test set, providing a measure of the model's broad multitask capabilities and general reasoning.

GSM8K The Grade School Math 8K (GSM8K) dataset (Cobbe et al., 2021) consists of 8.5K high-quality, linguistically diverse math word problems. These problems require multi-step reasoning using basic arithmetic operations. We use the 1,000 held-out test samples to evaluate mathematical reasoning performance via accuracy, measuring the model's ability to reach the correct numerical conclusion through interpretable natural language steps.

MBPP The Mostly Basic Python Problems (MBPP) dataset (Chen et al., 2021) contains 974 crowd-sourced programming tasks designed to be solvable by entry-level programmers. Each task includes a natural language description, a Python solution, and three automated test cases. We evaluate code generation proficiency using the sanitized test split via the pass@1 metric, ensuring the generated functions pass all functional test cases.

AlpacaEval AlpacaEval (Dubois et al., 2024) is an automatic benchmark for instruction-following language models, consisting of 805 instructions representative of real-world user interactions. While typically used with an LLM-based "win-rate" judge, we follow a more granular approach in this setting by scoring the reference completions using negative perplexity. This metric treats the human-written or GPT-4 reference responses as a target distribution, where higher values indicate a better alignment with the expected instruction-following behavior.

OpenMathInstruct-2 OpenMathInstruct-2 is a massive scale mathematical instruction tuning dataset containing approximately 14M question-solution pairs derived from 600K unique questions. It was constructed using Llama 3.1 models as teachers to generate high-quality synthetic reasoning chains. This dataset serves as the fine-tuning source for the `nvidia/OpenMath2-Llama3.1-8B` model used in our math setting.

D.2. Models

ViT-B/32 ViT-B/32 (Dosovitskiy et al., 2021) is a Vision Transformer model for image recognition that applies a pure transformer directly to sequences of image patches, rather than using convolutional layers. It was shown to perform strongly on transfer learning benchmarks when pre-trained at scale.

Llama-3.1-8B (Base) Llama-3.1-8B is a decoder-only transformer from Meta's Llama 3.1 family of foundation language models. We use the pretrained Hugging Face checkpoint `meta-llama/Llama-3.1-8B` as the common base model for all downstream merging runs.

OpenMath2-Llama3.1-8B (Math) `nvidia/OpenMath2-Llama3.1-8B` (Toshniwal et al., 2024) is a model obtained by fine-tuning Llama 3.1 8B on OpenMathInstruct-2. It is specifically optimized for mathematical reasoning and problem-solving through high-quality synthetic data.

DeepSeek-R1-Distill-Llama-8B (Code) `deepseek-ai/DeepSeek-R1-Distill-Llama-8B` (Guo et al., 2025) is a distilled Llama 3.1 8B checkpoint trained on reasoning data generated by DeepSeek-R1. It inherits advanced algorithmic logic and code generation capabilities from its larger teacher model.

Llama-3.1-8B-Instruct (Instruct) `meta-llama/Llama-3.1-8B-Instruct` is Meta's official instruction-tuned Llama 3.1 8B model. It is optimized for instruction-following quality and human-aligned dialogue through supervised fine-tuning and reinforcement learning.

D.3. Training

Vision Training Pipeline. For the vision experiments, we start from a pretrained ViT-B/32 CLIP backbone and fine-tune separate task-specific models on CIFAR-10, STL-10, Imagenette, and EuroSAT. The training pipeline uses CLIP normalisation, random resized crops and horizontal flips for training, and centre cropping for evaluation. We fine-tune for 15 epochs with AdamW, a learning rate of 5×10^{-5}, weight decay 10^{-4}, batch size 64, and a cosine annealing schedule.

LLM training pipeline. We start from the pretrained `Llama-3.1-8B` base model and merge three already fine-tuned task checkpoints: `nvidia/OpenMath2-Llama3.1-8B` for maths, `deepseek-ai/DeepSeek-R1-Distill-Llama-8B` for code, and `meta-llama/Llama-3.1-8B-Instruct` for instruction following. For each task, we construct a calibration set of prompt-only examples (30 per task in fast mode, 100 in full mode) and evaluate the merged models using teacher-forced MSE and cross-entropy on prompt-completion pairs, together with GSM8K exact-match accuracy, MBPP pass@1, and IFEval strict accuracy as task-level metrics.

QP construction and optimisation. To keep the residual updates small and consistent with the linearisation used in the theory, we update only the penultimate projection layer and the classification head during fine-tuning. For calibration, we use 100 examples per task and construct the quadratic objective from the task-specific base model, the task-specific head, and the Jacobian of the downstream normalisation-plus-head map at the merge-layer representation. In the implementation, the resulting QP is solved approximately by iterative gradient descent for 500 steps, initialised from model soup, rather than by an exact solver. This gives a much cheaper optimisation procedure in the high-dimensional coefficient space while preserving the behaviour needed for the merge experiments.

Merge strategies and optimisation. We compute task deltas relative to the base model and solve the diagonal QP by Adam on the merge coefficients, clamping them to $[0, 1]$. In the full setting, the QP is applied to every linear layer with 1,000 Adam steps in fast mode (3,000 in full mode); in the penultimate-layer setting, it is restricted to `model.layers.30.mlp.down_proj`; and in the sequential setting, the QP is applied layer by layer, updating the model after each layer so that subsequent layers optimise over the partially merged activations. For comparison, we also evaluate model soup, task arithmetic, DARE, and Fisher-weighted merging on the same deltas and calibration data.

D.4. Extra Results

We begin with a CLIP pretrained ViT-B/32 base model (Dosovitskiy et al., 2020; Radford et al., 2021) and fine-tune only the the final transformer block's MLP output projection, keeping all other backbone weights frozen. We finetune on four diverse downstream datasets (CIFAR-10, STL-10, Imagenette, and EuroSAT) to obtain four task-specific models. We then merge these models and evaluate performance using MSE and accuracy on each dataset individually, as well as on the combined dataset. In all experiments, the resulting QP is solved approximately using 100 steps of gradient descent, which is substantially cheaper than an exact solve and works well in the overparameterised setting. For the QP construction, we linearise the task-specific logit map around the calibration examples by computing the Jacobian of the downstream normalisation and head with respect to the representation at the merge layer, and use this local linearisation to form the output-space quadratic objective. For calibration, we use 100 examples per task to construct the quadratic objective; larger calibration sets may further improve performance.

D.5. ViT-32 Single-Layer

D.6. ViT-32 Multi-Layer

We next extend the setup to sequential multi-layer merging over both linear projections in the final transformer block MLP, fine-tuning these together with the task head. Following Appendix F, we solve the QP sequentially across the two layers while accounting for the residual connection in block 11. At each step, the Jacobian is recomputed from the current partially-merged model: for `fc1` it includes the path through activation, `fc2`, the residual addition, normalisation, and head; for `fc2` it includes the residual addition, normalisation, and head only. As shown in Figure 4, QP outperforms the

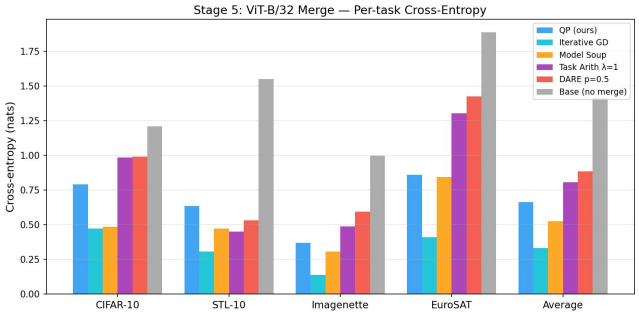
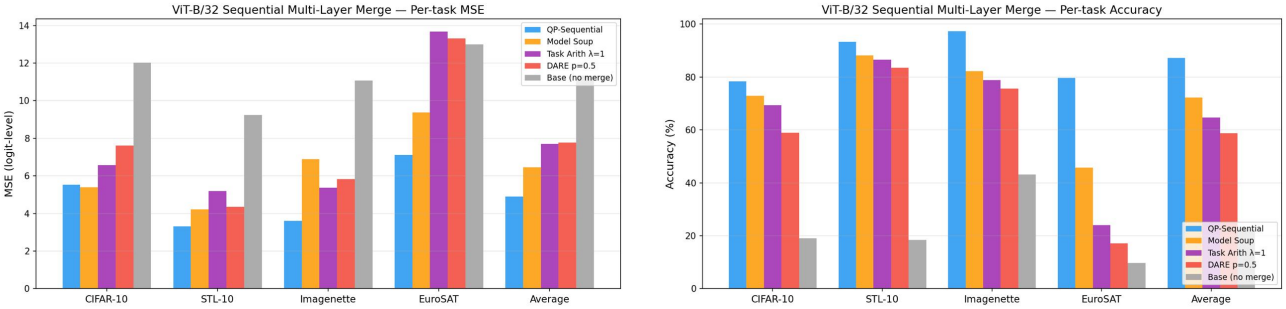


Figure 3. Vit-32 cross-entropy across benchmarks.



(a) MSE on ViT-B/32. *(b)* Accuracy on ViT-B/32.

Figure 4. Performance on ViT-32 across sequential multi-layer merging.

benchmark methods in both MSE and accuracy across all datasets, while the remaining methods perform substantially worse.

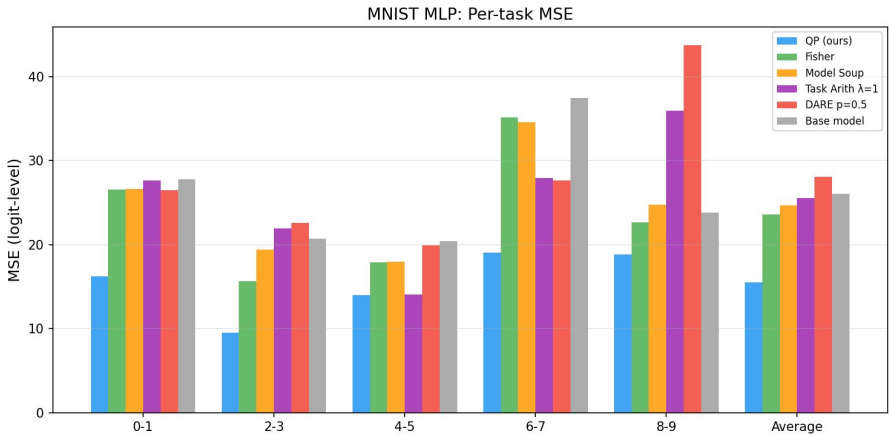
D.7. MNIST MLP

3-layer MLP with ReLU activations trained on MNIST, split into $K = 5$ non-overlapping digit-pair tasks: $\{0\text{-}1, 2\text{-}3, 4\text{-}5, 6\text{-}7, 8\text{-}9\}$. The base model is a $\text{Linear}(784, 256) \to \text{ReLU} \to \text{Linear}(256, 128) \to \text{ReLU} \to \text{Linear}(128, 10)$ network pretrained on all digits, with each task-specific model fine-tuned only on its two-digit subset. Merging is performed at `layer2` (the $256 \to 128$ linear layer), so the downstream network L consists of a ReLU nonlinearity followed by the final linear classifier—exercising the full linearisation path of our method.

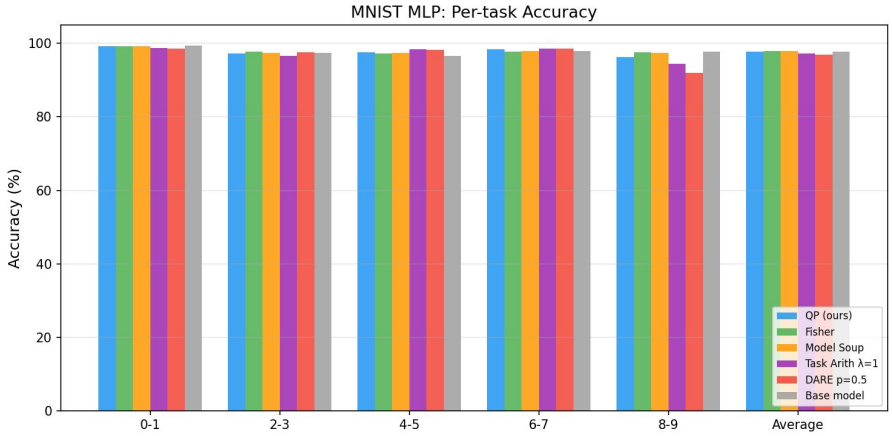
The QP is built from $n = 100$ calibration samples per task ($n_{\text{total}} = 500$), with targets set to the fine-tuned model's logits. The learned per-row blending weights $\mathbf{d}^\star \in [0, 1]^{K \times r}$ ($r = 128$) are optimised via Adam projected gradient descent for 500 steps with learning rate 10^{-2}, initialised at the uniform soup solution.

Despite the nonlinearity introduced by the ReLU between `layer2` and the output, the first-order linearisation underlying our method provides a useful local approximation of how weight perturbations propagate to logits. The Jacobian L is computed via JVP evaluated at the base model's `layer2` pre-activations, and the resulting quadratic programme recovers a merged model that matches or improves upon the model soup in average accuracy across tasks, while reducing logit-level MSE relative to the per-task fine-tuned models.

Results for this MNIST MLP are shown in Figure 5, showing the MSE loss for the quadratic programme always beats other methods.



(a) Per-task MSE (logit-level) against fine-tuned model outputs on the held-out test set.

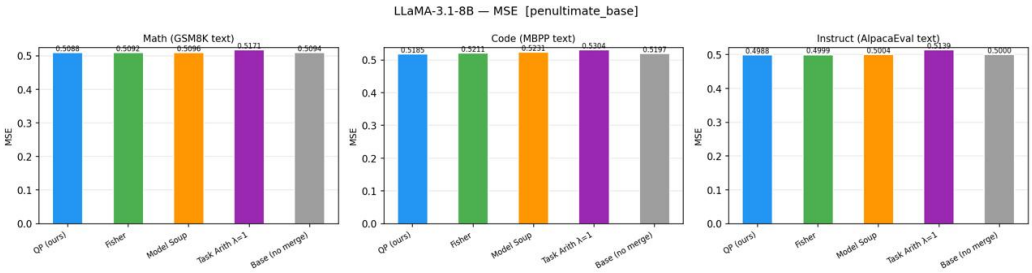


(b) Per-task classification accuracy on the held-out test set.

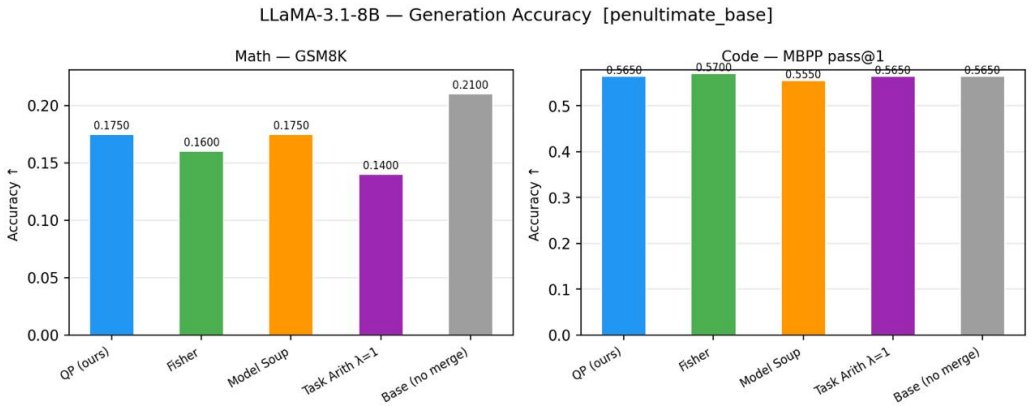
Figure 5. Performance of single-layer merging methods on a 3-layer MNIST MLP across five digit-pair tasks. Our QP method learns task-adaptive row-wise blending weights for `layer2`, while baselines (Model Soup, Task Arithmetic λ=1, DARE p=0.5, Fisher merging) apply fixed or heuristic mixing. The base model (no merging) is included as a lower bound.

D.8. LLaMA 3.1 Single Layer

Now we consider an LLM. We start from the pretrained Llama 3.1 8B base model and merge three task-specific finetunes for code, instruct, and maths. Details of the models and datasets are given in Appendix D.1. In the single-layer setting shown in Figure 6, QP attains the lowest MSE across the benchmarks, including relative to Fisher merging, although Fisher achieves higher code accuracy. This reflects the fact that lower MSE does not always correspond to better task accuracy, a limitation we discuss later.



(a) MSE on LLaMA 3.1.



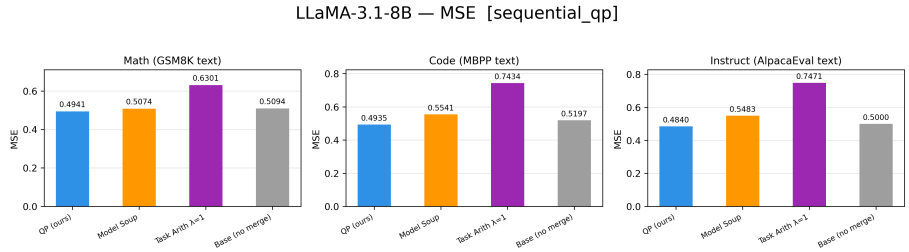
(b) Accuracy on LLaMA 3.1.

Figure 6. Performance on LLaMA 3.1 across penultimate-layer merging.

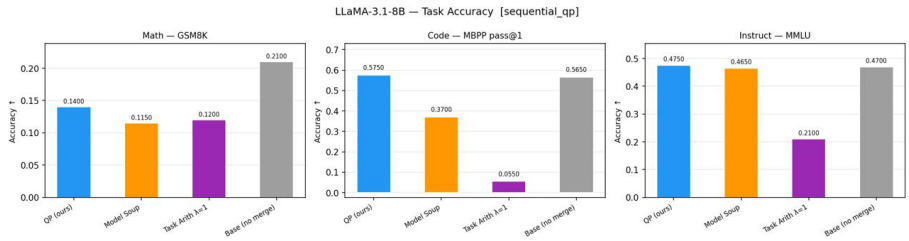
D.9. LLaMA 3.1 Multi-Layer Sequential

We extend this to multi-layer merging using the sequential scheme from Section F, merging all linear layers in the model. Results in Figure 7 show that the diagonal QP achieves the lowest MSE overall and the highest accuracy on two of the three tasks.

We present the cross-entropy results for the multi-layer sequential merge on LLaMA 3.1.



(a) MSE on LLaMA 3.1.



(b) Accuracy on LLaMA 3.1.

Figure 7. Performance on LLaMA 3.1 across sequential multi-layer merging.

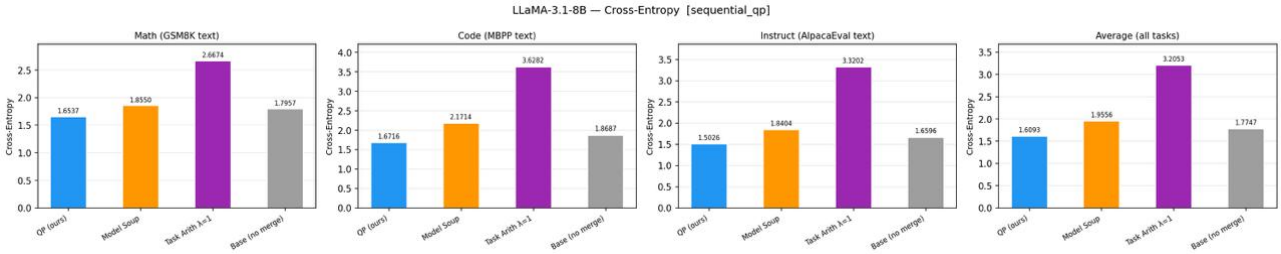


Figure 8. LLaMA 3.1 multi-layer sequential cross-entropy across benchmarks.

E. Merging in a General Basis

Having identified which output subspaces are optimal in the relaxed projection problem, we now turn to the realisable problem: given a fixed basis $\{q_p\}$ and the available residual updates $\{\delta_N^{(k)}\}$, what merge coefficients $\mathbf{d}$ minimise the calibration loss? The projection view characterises the best output directions for reducing residual error. We now derive the coefficient-tied optimisation problem that determines the optimal merge weights for a fixed basis and a fixed collection of model updates.

Recall that $P = \{q_1, \dots q_p\}$ is any orthonormal basis. For each basis direction q_p, model k, and sample j, define the projected activation as the amplitude of this sample at the residual layer N in direction q_p by:

$$\alpha_{k,p}^{(j)} \;:=\; q_p^\top \delta_N^{(k)} Z x_j \;\in\; \mathbb{R}, \tag{23}$$

and the projected residual as the target output propagated backwards to the residual layer N in direction Lq_p by:

$$\beta_p^{(j)} \;:=\; (Lq_p)^\top b_j, \qquad b_j := h(x_j;\theta) - y_j. \tag{24}$$

We parameterise the merged residual as a sum of rank-1 updates in the $\{q_p\}$ basis, with a scalar weight $d_{k,p} \in \mathbb{R}$ for model

k along direction q_p. The stacked weight vector is $\mathbf{d} = [d_{k,p}]_{k,p} \in \mathbb{R}^{KP}$. The merged output at sample j is

$$h(x_j; \theta + \mathbf{d}\boldsymbol{\delta}) \;=\; h(x_j;\theta) \;+\; L\left(\sum_{k=1}^{K}\sum_{p=1}^{P} d_{k,p}\, q_p\, q_p^\top\, \delta^{(k)}\right) Z x_j \tag{25}$$

$$= \; h(x_j;\theta) \;+\; \left(\sum_{k=1}^{K}\sum_{p=1}^{P} d_{k,p}\alpha_{k,p}^{(j)} L\, q_p\right), \tag{26}$$

noting that $d_{k,p}\alpha_{k,p}^{(j)}$ are scalars. Define $G \in \mathbb{R}^{P\times P}$ as encoding the coupling induced by the downstream map L:

$$G_{pp'} \;:=\; q_p^\top L^\top L\, q_{p'}. \tag{27}$$

Note that $L^\top L$ defines the metric under which directions interact and $G = I$ exactly when $L^\top L = I$ for any choice of orthonormal basis. In this instance, the directions decouple. The squared-output objective over the calibration set $\mathcal{K}$ is

$$\mathcal{J}(\mathbf{d}) = \sum_{j\in\mathcal{K}}\left[\sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{k'=1}^{K}\sum_{p'=1}^{P} d_{k,p}\, d_{k',p'}\, \alpha_{k,p}^{(j)}\, \alpha_{k',p'}^{(j)}\, G_{pp'} \;+\; 2\sum_{k=1}^{K}\sum_{p=1}^{P} d_{k,p}\, \alpha_{k,p}^{(j)}\, \beta_p^{(j)} \;+\; \|b_j\|^2\right].$$

Proposition E.1 (General-basis QP)**.** *Under the assumption that the model output is linear in the residual update, we obtain the following quadratic objective. Let*

$$\mathcal{J}(\mathbf{d}) = \tfrac{1}{2}\,\mathbf{d}^\top H\,\mathbf{d} + \mathbf{f}^\top\mathbf{d}$$

be the quadratic objective with indices flattened over $(k,p) \in \{1,\dots,K\}\times\{1,\dots,P\}$, where

$$H_{(k,p),(k',p')} \;=\; 2\sum_{j\in\mathcal{K}} \alpha_{k,p}^{(j)}\, \alpha_{k',p'}^{(j)}\, G_{pp'}, \qquad f_{(k,p)} \;=\; 2\sum_{j\in\mathcal{K}} \alpha_{k,p}^{(j)}\, \beta_p^{(j)},$$

and $G \in \mathbb{R}^{P\times P}$ is defined by $G_{pp'} = q_p^\top L^\top L q_{p'}$ (hence $G \succeq 0$) so that G captures the coupling between basis directions induced by the downstream map L. In the unconstrained case, we have the following:

1. *If $H \succ 0$, the minimiser is unique and given by $\mathbf{d}^* = -H^{-1}\mathbf{f}$.*
2. *If $H \succeq 0$ is singular and $\mathbf{f} \in \mathrm{Range}(H)$, then the set of minimisers is the affine subspace*

$$\{\, \mathbf{d}^* = -H^{+}\mathbf{f} + \mathbf{w} : \mathbf{w} \in \ker(H)\,\}.$$

The general-basis QP defines a single global optimisation over all merge coefficients. The Hessian H contains cross-terms between models and basis directions, reflecting the fact that their contributions to the output are coupled through shared inputs and the downstream map. As a result, the optimal coefficients cannot be determined independently, but must be solved jointly. Fixed or coordinate-independent rules such as task arithmetic ignore the off-diagonal coupling structure in H, while sequential layer-wise methods instead solve a sequence of restricted subproblems and are therefore greedy approximations to the joint objective. Our formulation instead solves the full coupled problem, yielding the minimum of the quadratic objective under the chosen basis.

When $G = I$ and $p = 1$, the solution reduces to the 1-D form in Proposition C.1.

E.1. Extension to Other Loss Functions

While our closed-form projection result is derived for squared loss, the notion of irreducible error extends to any convex loss as an optimal value gap induced by restricting the correction subspace. However, the resulting objective no longer admits a simple projection or trace-based characterisation.

Fisher-weighted merging (Matena and Raffel, 2022a) does not arise as an exact special case of our squared-output projection objective. Rather, it is a related parameter-space quadratic merging objective. Given fine-tuned checkpoints θ_k and diagonal Fisher matrices F_k, Fisher merging solves $\min_\theta \sum_k (\theta-\theta_k)^\top F_k(\theta-\theta_k)$, whose solution is $\theta^* = \left(\sum_k F_k\right)^{-1}\sum_k F_k\theta_k$. For classification models, the Fisher metric is induced by the likelihood, equivalently the negative log-likelihood or cross-entropy loss. Thus Fisher merging can be viewed as replacing the output-space squared-loss geometry used in our QP with a parameter-space Fisher metric; it fits our framework at the level of quadratic merge objectives, but not as an exact instance of the output-space projection formulation.

F. Multi-Layer Merging

Under the linear-in-update assumption, each subproblem of the single-layer-merge formulation remains a convex QP. In contrast, multi-layer merging introduces interactions between updates applied at different layers, and the objective is no longer convex. In particular, updates at different layers compose nonlinearly, so the optimal joint solution cannot be obtained by solving independent layer-wise problems. We therefore adopt practical approximations for the multi-layer setting.

Sequential layer-wise merging. A natural approach is to apply the single-layer QP sequentially across layers. At each step, the residual is updated using the current merged model, and the QP is solved for the next layer. This recomputes the upstream and downstream maps at each stage, so each subproblem remains an exact convex single-layer QP. However, this procedure is greedy: it produces a sequence of locally optimal updates without accounting for how earlier decisions affect later layers. As a result, sequential merging introduces an approximation error arising from cross-layer interactions. In particular, the effect of an update at one layer depends on updates applied at subsequent layers, which are not considered when solving earlier QPs. The interaction error arises from compositions of updates across layers. For example, in a two-layer setting where updates from two different fine-tuned models in distinct layers are δ_2 and δ_1, this term takes the form $\delta_2\delta_1$. More generally, the interaction error scales with products of layer-wise update norms and is second-order in the magnitude of the updates.

Hybrid merging with QP refinement. Alternatively, one can first obtain a multi-layer merged model using an existing method (e.g. model soups or task arithmetic), and then refine selected layers using the QP. This leverages the global structure captured by heuristic methods while using the QP to optimise layer-wise corrections in a principled way. In practice, both approaches trade off optimality for tractability in the multi-layer setting. Empirically, we find that controlling the magnitude of updates, for example via regularisation or by restricting merging to a subset of layers, improves performance, consistent with the role of cross-layer interactions.