When is Model Souping Tasty? Similarity, Transitivity and Robustness

Simon Ghyselincks [*,1,2] **Pierre Mackenzie** [*,1] **Evan Shelhamer** [1,2]

[1]University of British Columbia [2]Vector Institute

[*]*Equal contribution*

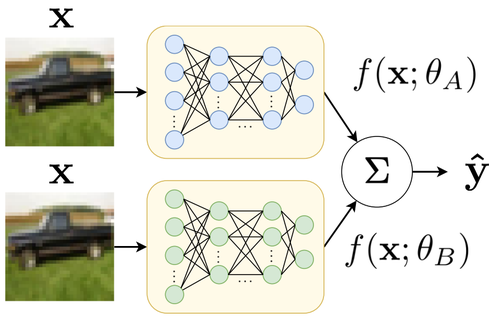
What Are Model Soups?

A **model soup** averages the parameters of independently fine-tuned models from a shared initialization:

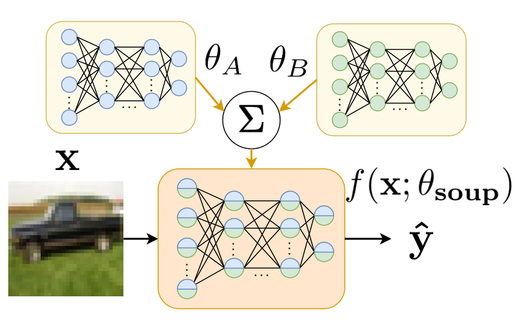
$$\theta_{\mathbf{soup}} = (1-\alpha)\theta_A + \alpha\theta_B$$

Unlike an **ensemble**, which averages the *outputs* of multiple forward passes, a soup produces a single model with **no added inference cost**. When ingredients lie in a shared low-loss basin, souping can outperform both parent models (Wortsman et al., 2022).

An **ensemble** is the combination of *outputs*



A **soup** is the combination of *parameters*



Why and When to Soup?

Adaptive: Interpolation weight α can be **tuned at test time** to trade off between ingredients specialized for different distributions (Croce et al., 2023).

Robust: Soups of models fine-tuned for different distributions can **generalize to unseen shifts** better than any single ingredient (Ramé et al., 2023).

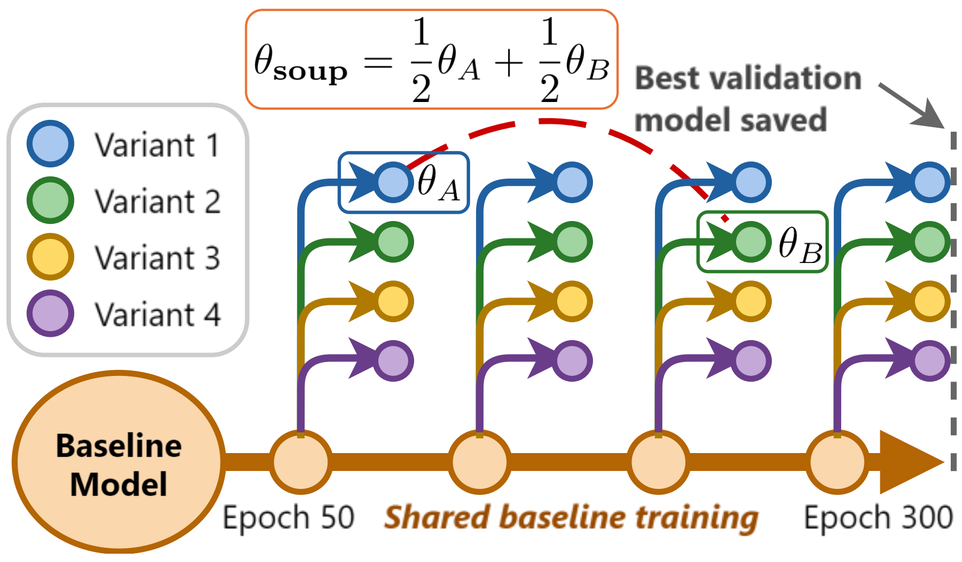
> *But not all ingredients soup well together, define:*
>
> $$\textbf{soup gain} = \mathrm{acc}(\theta_{\mathbf{soup}}) - \max\{\mathrm{acc}(\theta_A), \mathrm{acc}(\theta_B)\}$$
>
> *Where "acc" is classification accuracy over a test set.*

1. How much shared pre-training is needed? How does souping transition from incompatibility to performance?

2. Can similarity predict performance? Do standard model- parameter distance metrics forecast soup gain?

3. Is souping transitive? If A soups with B and B soups with C, does A soup with C?

Experimental Design



We train a ResNet-50 baseline on CIFAR-100, saving checkpoints every 10 epochs. From each of **26 branch points**, we fine-tune **4 variants** with different optimizer settings to convergence. This yields **104 models** and **5,356 pairwise soups** at the midpoint $\alpha = 0.5$.

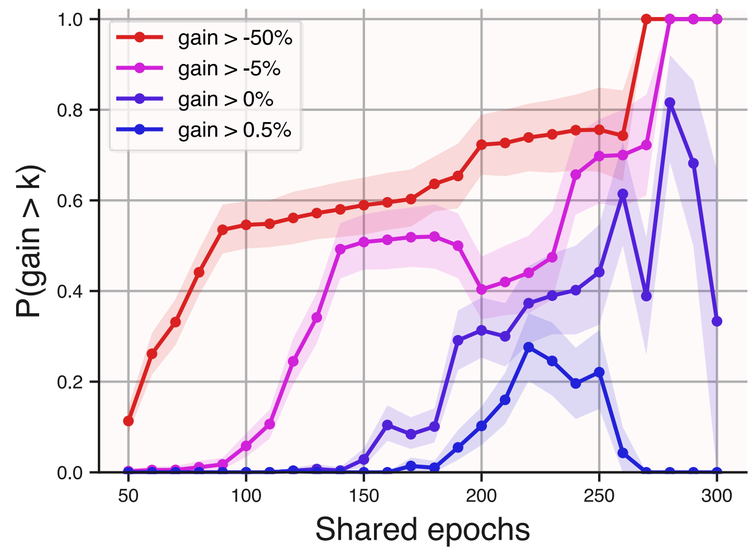
THE UNIVERSITY OF BRITISH COLUMBIA

VECTOR INSTITUTE | INSTITUT VECTEUR

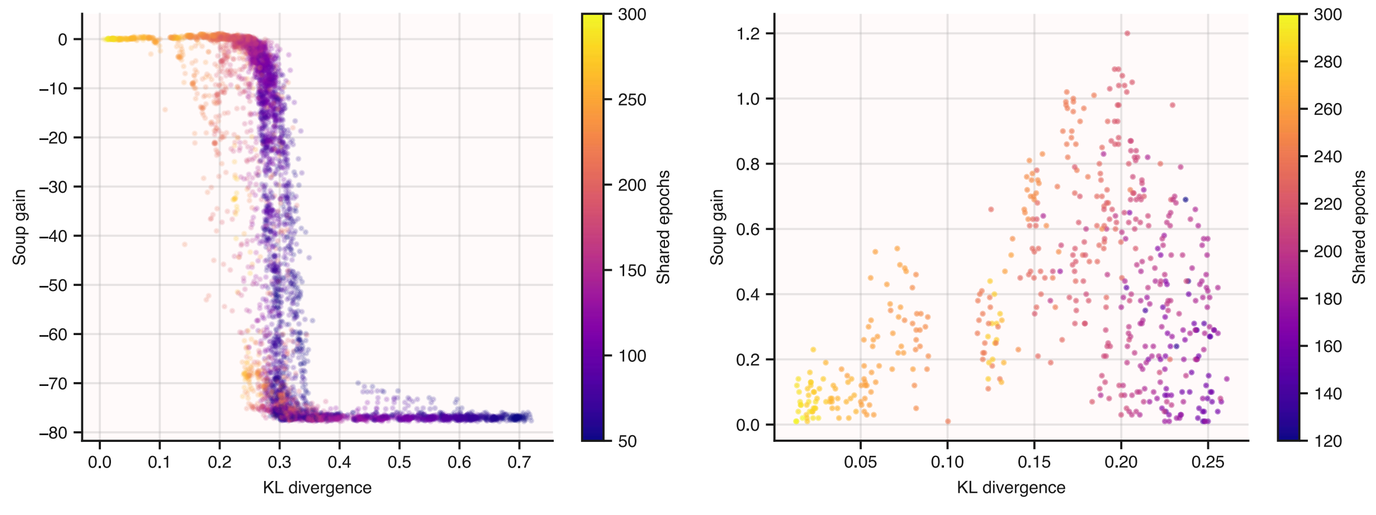
Code

Shared Training

As we increase the number of shared epochs, the probability of successful souping increases. However, to get the best performance boost, soups should not too many shared epochs either. **There is a tradeoff** between souping gain and risk, and a fine balance to be struck.



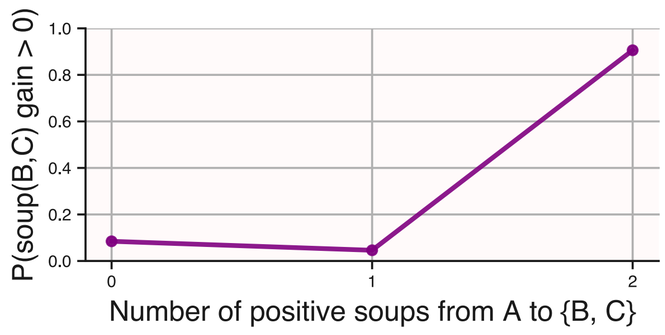
Similarity



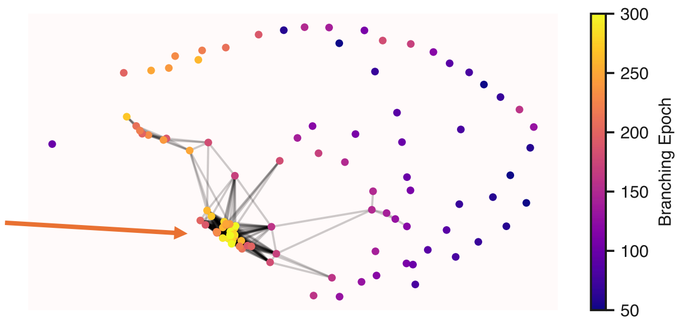
KL divergence is predictive of souping performance. Ingredients that don't soup well have high KL divergence. However, too little divergence and there is little to be gained. **A sweet spot** of similarity is required for the tastiest soups.

Transitivity

For there to be a high chance that B soups with C, A must soup with both B and C. **Souping is moderately transitive.**



We plot all ingredients using a distance metric based on souping. Edges denote positive soup gain. All edges can be found in a single connected component and **most triples are transitive**.



Souping To Go

We support the hypothesis that souping works due to optimised parameters sitting in the same low-loss basin, which is why transitivity usually holds. We also impart advice to ML practitioners: if you're looking for the **tastiest soups**, they should **use the right cooking time!**

Future work includes repeating with other models and datasets, seasoning our soups (Croce et al., 2023) and formalising a theory of souping in shallow networks.