Soups

What is soup?

Uniform soup

Just average all parameters

Greedy soup

Sort models by accuracy

Take new ingredient if soup starts to be better

Recipe 1 GreedySoup

Input: Potential soup ingredients $\{\theta_1, ..., \theta_k\}$ (sorted in decreasing order of ValAcc(θ_i)).
ingredients $\leftarrow \{\}$
for $i = 1$ **to** k **do**
  if ValAcc(average(ingredients $\cup \{\theta_i\}$)) $\geq$
      ValAcc(average(ingredients)) **then**
    ingredients $\leftarrow$ ingredients $\cup \{\theta_i\}$
return average(ingredients)

Learning soup

All models must be loaded in memory

Learn summarizing coefficients

$$\underset{\alpha \in \mathbb{R}^k, \beta \in \mathbb{R}}{\arg\min} \sum_{j=1}^{n} \ell\left(\beta \cdot f\left(x_j, \sum_{i=1}^{k} \alpha_i \theta_i\right), y_j\right).$$

Uniform soup problem

Bad models = bad soup

Experiments

Fine tuning end-to-end

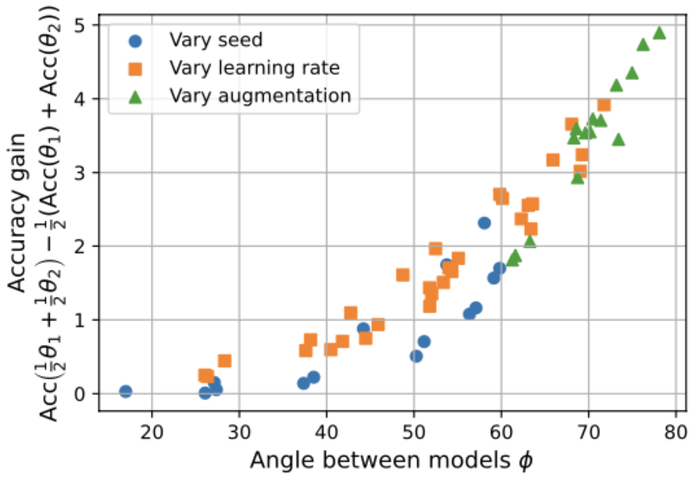

Figure 3: The advantage of averaging solutions (y-axis) is correlated with the angle ϕ between between solutions, while varying hyperparameter configurations between pairs enables a larger ϕ. Each point corresponds to a pair of models θ_1, θ_2 that are fine-tuned independently from a shared initialization θ_0 with different hyperparameter configurations. The angle ϕ between between solutions refers to the angle between $\theta_1 - \theta_0$ and $\theta_2 - \theta_0$ (i.e., the initialization is treated as the origin). Accuracy is averaged over ImageNet and the five distribution shifts described in Section 3.1.

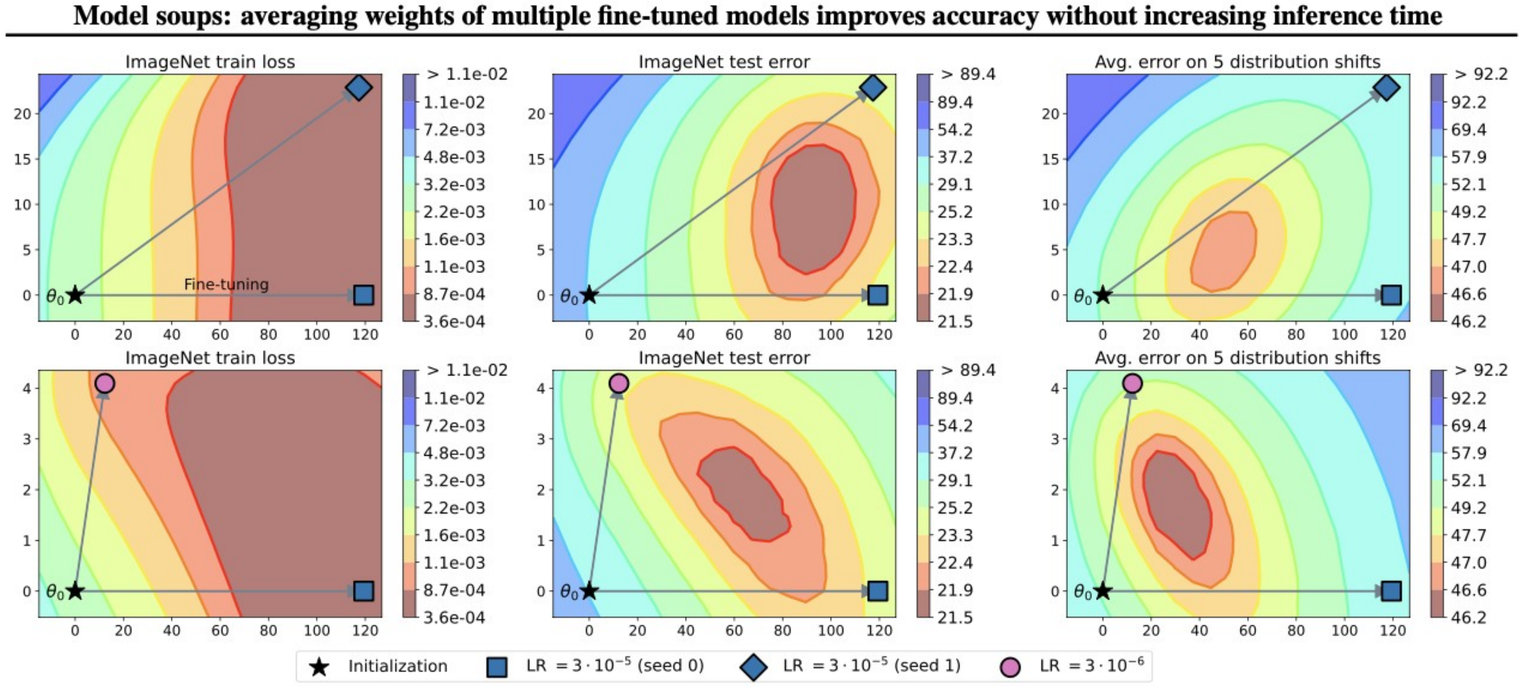

Figure 2: The solution with the highest accuracy is often not a fine-tuned model but rather lies between fine-tuned models. This figure shows loss and error on a two dimensional slice of the loss and error landscapes. We use the zero-shot initialization θ_0 and fine-tune twice (illustrated by the gray arrows), independently, to obtain solutions θ_1 and θ_2. As in Garipov et al. (2018), we obtain an orthonormal basis u_1, u_2 for the plane spanned by these models, and the x and y-axis show movement in parameter space in these directions, respectively.

Experiments

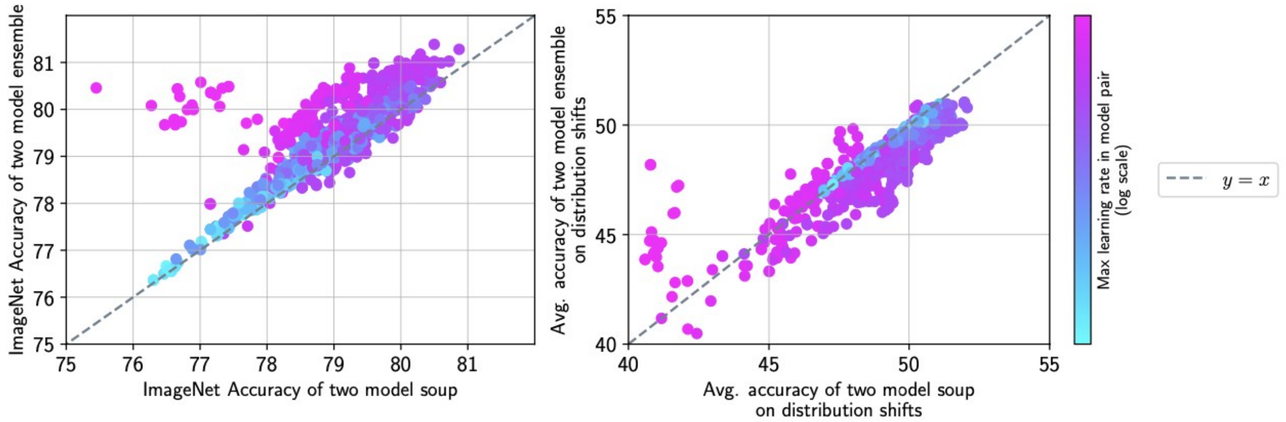

Figure 4: Ensemble performance is correlated with model soup performance. Each point on the scatter plot is a model pair with different hyperparameters. The x-axis is the accuracy when the weights of the two models are averaged (i.e., the two model soup) while the y-axis is the accuracy of the two model ensemble. Ensembles often perform slightly better than soups on ImageNet (left) while the reverse is true on the distribution shifts (right). Each model pair consists of two random greed diamonds from Figure 1.

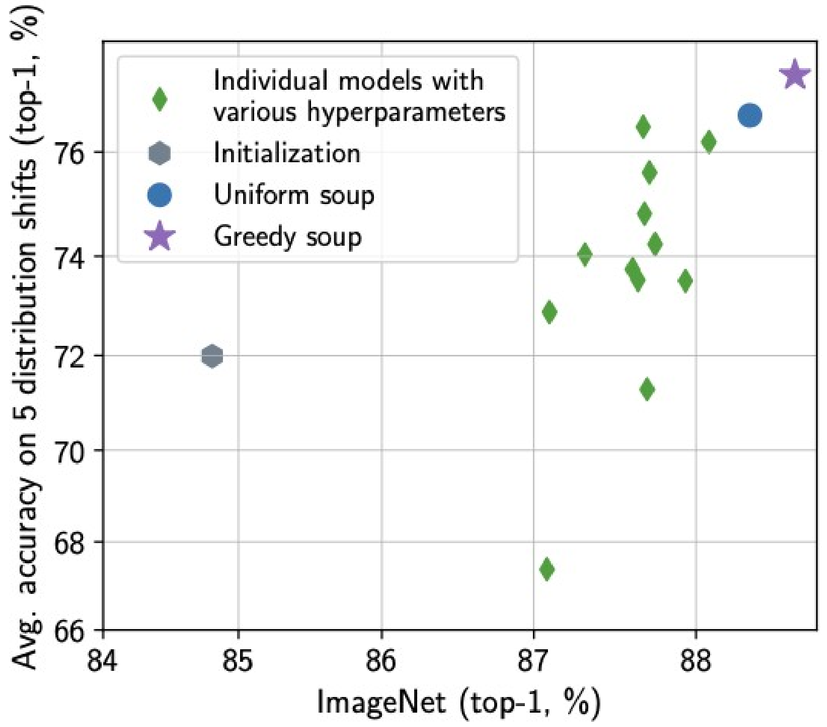

Figure 5: *Model soups* improve accuracy when fine-tuning ALIGN.

Experiments

Table 4: Greedy soup improves over the best individual models obtained in a hyperparameter sweep for ViT-G/14 pre-trained on JFT-3B and fine-tuned on ImageNet, both in- and out-of-distribution. Accuracy numbers not significantly different from the best are bold-faced. Statistical comparisons are performed using an exact McNemar test or permutation test at $\alpha = 0.05$. Avg shift accuracy of the best model on each test set is the best average accuracy of any individual model. Analogous results when fine-tuning BASIC-L are available in Appendix C.

	ImageNet			Distribution shifts					
Method	Top-1	ReaL	Multilabel	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts
ViT/G-14 (Zhai et al., 2021)	90.45	90.81	–	83.33	–	–	70.53	–	–
CoAtNet-7 (Dai et al., 2021)	90.88	–	–	–	–	–	–	–	–
Our models/evaluations based on ViT-G/14:									
ViT/G-14 (Zhai et al., 2021) (reevaluated)	90.47	90.86	96.89	83.39	94.38	72.37	71.16	89.00	82.06
Best model on held out val set	90.72	91.04	96.94	83.76	95.04	73.16	78.20	91.75	84.38
Best model on each test set (oracle)	90.78	**91.78**	**97.29**	**84.31**	95.04	73.73	**79.03**	92.16	84.68
Greedy ensemble	**90.93**	91.29	**97.23**	**84.14**	94.85	73.07	77.87	91.69	84.33
Greedy soup	**90.94**	91.20	**97.17**	**84.22**	**95.46**	**74.23**	78.52	**92.67**	**85.02**

Table 5: Performance of model soups on four text classification datasets from the GLUE benchmark (Wang et al., 2018).

Model	Method	MRPC	RTE	CoLA	SST-2
BERT (Devlin et al., 2019b)	Best individual model	88.3	61.0	59.1	92.5
	Greedy soup	88.3 (+0.0)	61.7 (+0.7)	59.1 (+0.0)	93.0 (+0.5)
T5 (Raffel et al., 2020b)	Best individual model	91.8	78.3	58.8	94.6
	Greedy soup	92.4 (+0.6)	79.1 (+0.8)	60.2 (+0.4)	94.7 (+0.1)

	ImageNet	Dist. shifts
Best individual model	80.38	47.83
Second best model	79.89	43.87
Uniform soup	79.97	51.45
Greedy soup	81.03	50.75
Greedy soup (random order)	80.79 (0.05)	51.30 (0.16)
Learned soup	80.89	51.07
Learned soup (by layer)	81.37	50.87
Ensemble	81.19	50.77
Greedy ensemble	81.90	49.44

Why it works?

We derive the following approximation for the loss difference:

$$\mathcal{L}_\alpha^{\text{soup}} - \mathcal{L}_\alpha^{\text{ens}} \approx \frac{\alpha(1-\alpha)}{2}\Bigg(-\frac{\mathrm{d}^2}{\mathrm{d}\alpha^2}\mathcal{L}_\alpha^{\text{soup}} + \beta^2 \mathbb{E}_x \mathrm{Var}_{Y\sim p_{\text{sftmx}}(\beta f(x;\theta_\alpha))}\left[\Delta f_Y(x)\right]\Bigg), \qquad (1)$$

where $[p_{\text{sftmx}}(f)]_i = e^{f_i}/\sum_j e^{f_j}$ is the standard "softmax" distribution and $\Delta f(x) = f(x;\theta_1) - f(x;\theta_0)$ is the difference between the endpoint logits. We obtain our approximation in the regime where the logits are not too far from linear; see Appendix K.3 for a detailed derivation.

The first term in approximation (1) is negatively proportional to the second derivative of the loss along the trajectory: when the approximation holds, convexity of the loss indeed favors the soup. However, the second term in the approximation does not follow from the "convex basin" intuition. This term always favors the ensemble, but is small in one of two cases: (a) the somewhat trivial case when the endpoint models are similar (so that Δf is small) and (b) when the soup produces confident predictions, implying that $p_{\text{sftmx}}(\beta f(x;\theta_\alpha))$ is close to a point mass and consequently the variance term is small.