

Souper-Model: How Simple Arithmetic Unlocks State-of-the-Art LLM Performance

Shalini Maiti^{1,3,†}, Amar Budhiraja^{2,†}, Bhavul Gauri², Gaurav Chaurasia²,
Anton Protopopov², Alexis Audran-Reiss², Michael Slater², Despoina Magka²,
Tatiana Shavrina², Roberta Raileanu^{2,3,*}, Yoram Bachrach²

¹Meta Superintelligence Labs, ²FAIR at Meta, ³University College London

† Equal contribution * Work done while working at Meta

Abstract

Large Language Models (LLMs) have displayed remarkable capabilities across diverse domains, but their training remains resource- and time-intensive, requiring massive computational resources and careful orchestration of training procedures. Model souping—the practice of averaging weights from multiple models of the same architecture—has emerged as a promising pre- and post-training technique that can enhance performance without expensive retraining. We observe that previous souping approaches can lead to collapse in precision-sensitive LLM capabilities. In this paper, we introduce *Soup Of Category Experts* (SoCE), a principled approach for model souping to overcome this shortcoming. The proposed method utilizes benchmark composition to identify optimal model candidates and applies non-uniform weighted averaging to maximize performance. Contrary to previous approaches, our method leverages the observation that different clusters (or categories) of points within a benchmark often exhibit low inter-correlations in model performance. SoCE identifies "expert" models for each weakly-correlated category cluster and combines them using optimized weighted averaging rather than uniform weights. We demonstrate that SoCE improves performance and robustness across multiple domains and achieves state-of-the-art results on the Berkeley Function Calling Leaderboard.

1 Introduction

Large Language Models (LLMs) have emerged as transformative technologies, demonstrating remarkable capabilities across diverse domains from natural language understanding to code generation and tool use (Brown et al., 2020; Comanici et al., 2025; Grattafiori et al., 2024; Rozière et al., 2024; Schick et al., 2023; Touvron et al., 2023a,b; Achiam et al., 2023; Team et al., 2025). The training paradigms for these foundational models typically involve

massive-scale pretraining on diverse corpora followed by supervised fine-tuning and reinforcement learning from human feedback (Ouyang et al., 2022; Bai et al., 2022). However, current training procedures remain extremely resource and time-intensive, often requiring large amounts of compute power and careful orchestration of training data mixtures to achieve desired competencies in the models (Hoffmann et al., 2022; Touvron et al., 2023b; Zhang et al., 2022).

Recent advances in model souping—averaging weights from multiple models of the same architecture—have shown promising results as a post-training technique to enhance performance without requiring expensive retraining. Prior work has demonstrated that uniform averaging of model weights can lead to improved performance across various tasks (Wortsman et al., 2022a) and mitigate issues like catastrophic forgetting (Kleiman et al., 2025). These findings suggest that model souping offers a computationally efficient alternative to traditional approaches that rely on modifying training data recipes or retraining procedures. Recently, the concept of "model souping" has evolved into a broader model merging paradigm that combines the weights of multiple task-specific derivative models into a single, highly capable network without the need for additional training. TIES-Merging (Yadav et al., 2023), an example of such an approach, isolates the fine-tuning updates and resets up to 80% of the smallest parameter changes to zero, retaining only the most significant delta weights. DARE (Yu et al., 2024) proposes randomly dropping a vast majority of these fine-tuned delta weights and rescaling the remaining parameters to maintain their overall mathematical expectation. While these approaches have been shown to push performance, our experiments show that these approaches can collapse several LLM capabilities which require exact instruction adherence, such as tool-use, long-context, and reasoning (Tables 3, 6

and 8). To address these concerns, we introduce an improved dense model averaging method that combines benchmark-guided candidate selection with non-uniform weighting to form a model soup leading to state-of-the-art on the Berkeley Function Calling Leaderboard (BFCL) (Patil et al., 2025). The contributions of this paper are as follows:

1. **Benchmark Aware Souping:** We introduce *Soup Of Category Experts* (SoCE), a novel model souping technique that leverages benchmark composition through category-aware expert selection mechanism. Unlike previous approaches, SoCE doesn’t lead to collapse of precision-sensitive capabilities like tool-use.
2. **State-of-the-Art Performance:** We demonstrate the efficiency of the proposed method across diverse domains, including state-of-the-art results for the BFCL (Patil et al., 2025). Our approach outperforms existing baselines, validating the effectiveness of category-specific model souping.
3. **Improved Model Consistency:** We perform a large-scale empirical analysis to show that model souping enhances performance consistency across benchmark categories. Souped models exhibit significantly higher Pearson correlations between category performances within a benchmark across model populations compared to their unsouped counterparts, indicating improved robustness and coherence across diverse task types.

We also discuss the connection of model souping with game theory and Shapley values; revisit the assumptions of existing model merging approaches in the context of specialized models; present qualitative analysis for BFCL experiments; understand impact on other benchmarks when souping for a given benchmark and present additional results on FLORES (Goyal et al., 2021) multi-lingual benchmark as part of appendix to this paper.

2 Related Work

We outline the current souping/merging methodologies that have emerged to improve LLM efficiency and generalization to contextualize SoCE’s novelty and usefulness within this space.

Souping: Starting with Utans (1996), model Souping has been explored in the machine learning literature in multiple contexts (Jang et al., 2025; Jaiswal et al., 2023; Kleiman et al., 2025; Wortsman et al., 2022a; Yu et al., 2024; Zimmer et al.,

2024). Weight averaging found theoretical foundation with (Izmailov et al., 2018), that showed that averaging checkpoints along a stochastic gradient descent trajectory finds flatter, better-generalizing minima; Wortsman et al. (2022a) extended this idea across independently fine-tuned models, showing how souping on model variants across different hyperparameter configurations of the same base model leads to improvements over individual candidates. They proposed 3 strategies: Uniform (where all models are weighted equally), Greedy (add 1 model at a time in decreasing order of performance) and Learned-Souping (soup weights are learned via gradient descent), with Greedy Souping yielding the most promising results. Wortsman et al. (2022b) further showed that interpolating between zero-shot and fine-tuned weights improves robustness under distribution shift, establishing weight-space interpolation as a practical regularization. Jang et al. (2025) improve upon the work by Wortsman et al. (2022a) using a geometric insight - proximity to pre-trained models - to reduce the number of models to two to achieve better results than its predecessor. SoCE takes it further by utilizing the insights from the correlation of metrics within a benchmark and co-operative game theory to formalize the selection of the best candidates and their souping proportion. The viability of weight averaging rests on the theoretical observation that models fine-tuned from the same pretrained weights remain in the same loss basin (Neyshabur et al., 2020) and exhibit linear mode connectivity (Frankle et al., 2020) - preconditions naturally satisfied by all candidate models in SoCE, which share a common pretrained initialization. Ramé et al. (2022) further provided a formal bias-variance-covariance decomposition showing that weight averaging succeeds when model diversity is high, reducing covariance between model errors - the condition SoCE engineers by selecting anti-correlated category experts. In a related study (Yu et al., 2024), the authors take several homologous (same architecture and parameters) models and perform deactivation and reactivation of neurons and merge the newly derived models to achieve better models compared to the input candidates. Model souping has also been used in continual learning to address catastrophic forgetting during finetuning of LLMs (Kleiman et al., 2025), and in RLHF (Ramé et al., 2024) where weight averaging of reward models has been shown to improve alignment. In LLMs (Grattafiori et al., 2024; Cohere et al., 2025), it has been used

Property	SoCE (Proposed Approach)	TIES (Yadav et al., 2023)	DARE (Yu et al., 2024)	Model Soups (Wortsman et al., 2022a)	Model Stock (Jang et al., 2025)
Operates on	Raw LLM weights	Task vectors	Task vectors (preprocessor)	Raw weights	Raw weights (layer-wise)
Candidate selection	Yes (correlation-guided)	No	No	No (all or greedy on val)	No (fixed: 2 models)
Weight optimisation	Non-uniform grid search	Majority vote on signs	Rescale by $1/(1-p)$	Uniform or greedy	Closed-form (angle θ)
Data needed at merge	Benchmark eval scores	None	None	Held-out val set (greedy)	None
Modifies parameters	Yes (weighted avg)	Yes (trim + sign elect)	Yes (random zero + rescale)	Yes (averaged weights)	Yes (weighted avg, layer-wise)
Training required	No	No	No	No	No (periodic merging optional)
Interference handling	Complementary expert selection	Trim + sign resolution	Sparsification of deltas	None (relies on shared basin)	Geometric anchoring to pretrained
Formal guarantee	No	No	Yes (in expectation)	Partial	Yes (given angular consistency)
Scalability with N	Scales (selects subset)	Better than TA as N grows	Plug-in, scales with host	Scales (greedy filters)	Designed for $N = 2$; extends to N
Risk to critical neurons (Qin et al., 2026)	Low (conservative avg)	Higher (trim discards)	Higher (random zeroing)	Low (conservative avg)	Low (pretrained anchor preserves)
Models needed	Subset of candidate pool	≥ 2	≥ 2 (preprocessor)	≥ 2 (typically many)	Exactly 2
Domain	(V)LLM (Complex benchmarks)	Vision + NLP	NLP (LLM-focused)	Vision + NLP	Vision (CLIP)
Pretrained model used	No	As origin (implicit)	As origin (implicit)	No	Yes (explicit geometric anchor)
Category structure	by within-benchmark	diverse tasks	diverse tasks	diverse tasks	diverse tasks

Table 1: Comparison of SoCE and other Model Merging methods. Refer to Table 7 for more methods.

as part of the fine-tuning pipeline. A principal distinction between all prior souping approaches and the method introduced here lies in their respective objectives: previous work has primarily leveraged souping to achieve inter-benchmark improvements - combining models specialised in coding and function calling (Cohere et al., 2025) or across 16 specialised domains (Abdin et al., 2025) - whereas SoCE systematically applies souping to enhance performance on a single benchmark by exploiting its internal compositional structure through Pearson correlation analysis of subcategory scores, encapsulating capability complementarity directly into the candidate selection process itself.

Automatic model merging: Several works (Matena and Raffel, 2022; Stoica et al., 2024; Yadav et al., 2023; Abdin et al., 2025) explore automatic model merging techniques. Ilharco et al. (2023) introduced task arithmetic, defining task vectors as the difference between fine-tuned and pretrained weights and showing these vectors can be added, negated, or combined to edit model behaviour - however, this suffers from interference when models push shared parameters in conflicting directions. Yadav et al. (2023) proposed TIES-Merging to address this - redundant low-magnitude parameters and sign conflicts - and resolving them through a trim-elect-merge procedure, while Yu et al. (2024) introduced

DARE, demonstrating that 90-99% of delta parameters can be randomly dropped and rescaled without degrading performance, substantially reducing interference before merging. Matena and Raffel (2022) proposed Fisher-weighted averaging using the Fisher information matrix to assign per-parameter importance weights, and Yang et al. (2023) introduced AdaMerging, which discovers optimal layer-wise merging coefficients via entropy minimization without requiring labeled data, with an application to classification and ViT models. Jin et al. (2022) developed RegMean, a closed-form merging solution via inner product regression. Akiba et al. (2025) apply evolutionary algorithms to model merging to find high-performing multi-modal LLMs. We provide a tabular summary of the properties of SoCE versus other merging methods in Tables 1, 7. Existing merging methods share certain characteristics that make them difficult to apply to current LLMs: requiring either privileged access to training data or activation statistics (Jin et al., 2022), impose architectural constraints (Jang et al., 2024), degrade as the number of merged models grows (Ilharco et al., 2023), or risk destroying catastrophically critical neurons identified by Qin et al. (2026) through aggressive parameter trimming/random zeroing (Yadav et al., 2023; Yu et al., 2024). Even Jang et al. (2025) and Wortsman et al. (2022a)

select candidates based purely on overall validation accuracy rather than structured capability coverage - whereas SoCE uniquely positions itself to avoid such failure modes by operating entirely in benchmark performance space, requiring only publicly available evaluation scores, and preserving all weights through conservative weighted averaging that minimizes critical neuron disruption.

3 Methodology: *Soup Of Category Experts*

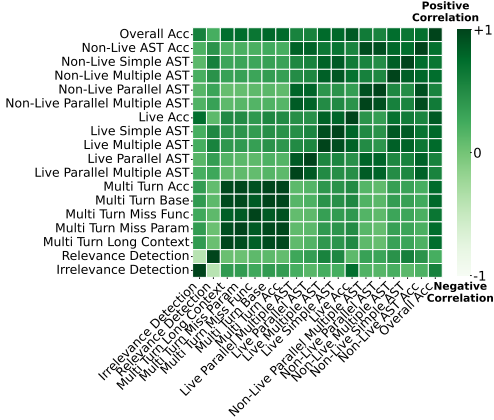


Figure 1: Pearson Correlation of models' performance on BFCLv3.

In this section, we present our method for model selection and weighting the selected models with the benchmark composition in mind.

The fundamental insight underlying our approach is that benchmark performance across categories exhibits heterogeneous correlation patterns. Different models demonstrate varying expertise across benchmark categories, with some categories being strongly correlated while others remain weakly correlated or even negatively correlated in terms of cross-model performance. To illustrate this phenomenon, we analyze the Berkeley Function Calling Leaderboard (BFCLv3) (Patil et al., 2025), which comprises multiple categories including multi-turn function calling, irrelevance detection, and function calling across different programming languages (Java, Javascript, etc.). Figure 1 presents a correlation heatmap showing Pearson correlation coefficients between category performances across all models on the leaderboard, where darker regions indicate higher correlations. The heatmap reveals strong positive correlations (dark green regions) between related categories and weak to negative correlations (light green regions) between unrelated categories. For instance, multi-turn categories exhibit high inter-correlation

(0.96-0.98), indicating that models proficient in one multi-turn task typically excel across all multi-turn scenarios. Conversely, weak correlation (0.07) exists between Multi-turn-base (where the model is evaluated on Multi-turn function calling aspect) and Live Accuracy (where the model is evaluated on real-world user collected function calling prompts) categories, suggesting these represent distinct competency domains.

Algorithm 1 *Soup Of Category Experts* (SoCE)

Input: Benchmark dataset $\mathcal{D}$ with categories $\{C_1, C_2, \dots, C_k\}$

Input: Set of candidate models $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$

Input: Correlation threshold τ for identifying low-correlation categories

Output: Souped model M_{soup}

- 1: **Step 1: Analysis**
 - 2: **for** each pair of categories (C_i, C_j) where $i \neq j$ **do**
 - 3: Compute Pearson correlation $\rho_{i,j}$ between $(P_1^i, \dots, P_n^i)$ and $(P_1^j, \dots, P_n^j)$ across all models and category performances
 - 4: **end for**
 - 5: $L = \{C_k \mid \exists l \text{ such that } |\rho_{k,l}| < \tau \text{ or } |\rho_{l,k}| < \tau\}$
 - 6: **Step 2: Expert Model Selection**
 - 7: **for** each category $C_i \in L$ **do**
 - 8: **for** each model $M_j \in \mathcal{M}$ **do**
 - 9: $P_j^i = \text{Performance}(M_j, C_i)$
 - 10: **end for**
 - 11: Select expert model $M_i^* = \arg \max_{M_j \in \mathcal{M}} P_j^i$
 - 12: **end for**
 - 13: **Step 3: Weight Optimization**
 - 14: Apply uniform grid search to find optimal weights, such that for $\mathbf{w} = \{w_1, \dots, w_l\}$ it is $\sum_{i=1}^l w_i = 1$, $l = |L|$ and $\mathbf{w}^* = \arg \max_{\mathbf{w}} \frac{1}{k} \sum_{i=1}^k \text{Performance} \left(\sum_{j=1}^l w_j \cdot M_j^*, C_i \right)$
 - 15: **Step 4: Souped Model**
 - 16: Create souped model: $M_{\text{soup}} = \sum_{i=1}^l w_i^* \cdot M_i^*$
 - 17: **return** M_{soup}
-

Our proposed method, *Soup Of Category Experts* (SoCE), exploits these correlation patterns to strategically select and weight models for souping. The core principle is to identify expert models for each weakly-correlated category cluster and aggregate them using optimized weighted averaging to combine complementary expertise. Algorithm 1 formalizes this procedure, which comprises four key steps: (1) correlation analysis to identify weakly-correlated category pairs, (2) expert model selection for each category based on performance rankings, (3) weight optimization to maximize aggregate performance, and (4) weighted model souping to produce the final combined model. For weight optimization, we perform a search over a uniform set of weights. We iterate over all combinations in the weight space with a maximum weight value of 0.9, lowest weight value of 0.1 and step size of 0.1,

where each weight corresponds to a single model . We also add a special case of equal weighing of the candidates to compare uniform souping. We discuss the implications of the correlation threshold (τ) on the proposed approach in Section A.6.

4 Experiments

4.1 Benchmarks and Baselines

We conduct comprehensive experiments to evaluate the effectiveness of our proposed *Soup Of Category Experts* (SoCE) methodology across three diverse benchmarks that span different LLM capabilities:

1. **Berkeley Function Calling Leaderboard (BFCL)** (Patil et al., 2025): evaluates tool calling and function invocation capabilities of LLMs across multiple categories including multi-turn interactions, irrelevance detection, and cross-language function calling. We use BFCL-v3 commit id 1bb65c9 for evaluations as stated in the public BFCL-v3 leaderboard.
2. **Multilingual Grade School Math Benchmark (MGSM)** (Shi et al., 2022): assesses mathematical reasoning abilities across multiple languages, testing both computational skills and cross-lingual generalization.
3. **∞ Bench** (Zhang et al., 2024): evaluates long-context processing capabilities, testing models’ ability to maintain coherence and extract information from extended sequences. We use a subset containing 3 categories.

For all benchmarks, we compare individual models with the following model soups:

1. **Uniform Souping (All Candidate Models)** (Wortsman et al., 2022a): Baseline approach uniformly averaging all candidate models.
2. **Uniform Soup with SoCE Model Selection**: Uniform weighting applied to our selected candidates based on anti-correlated categories.
3. **SoCE (Weighted Souping with Model Selection)**: Complete proposed method with candidate selection and optimized weighting.

This experimental design enables targeted ablation studies: comparing (1) vs (2) isolates our candidate selection benefits, (2) vs (3) quantifies weight optimization gains, and (1) vs (3) demonstrates overall SoCE performance improvements. We also compare current state-of-the-art individual models to

quantify improvements of souping over these models. We also evaluate SoCE against TIES (Yadav et al., 2023), DARE (DARE-Linear) (Yu et al., 2024) and DARE-TIES (Yu et al., 2024) on BFCL.

4.2 Numerical Takeaways

We compare souping for two model lines on BFCL: 70 billion and 8 billion parameter dense models. We identified 4 70B model candidates from the official leaderboard and applied SoCE to them. SoCE achieved 80.68% accuracy, establishing a new state-of-the-art with a 2.84% improvement over the previous best-performing individual model, xLAM-2-70b-fc-r (Prabhakar et al., 2025) (78.45%). The optimal configuration utilized xLAM-2-70b-fc-r, CoALM-70B (Acikgoz et al., 2025), and watt-tool-70B (gm8.ai, 2024a) with 0.5, 0.2 and 0.3 weights, respectively. For 8B models, SoCE achieved 76.50% accuracy, surpassing the previous state-of-the-art within the 8B model size, xLAM-2-8b-fc-r (Prabhakar et al., 2025) by 5.7% relative, with optimal weights of 0.7, 0.2, and 0.1 for xLAM-2-8b-fc-r, ToolACE-2-8B (Liu et al., 2024), and watt-tool-8B (gm8.ai, 2024b) respectively. For ablation studies on SoCE’s model selection, we compare Uniform Souping (all candidate models) with Uniform Souping with SoCE model selection, and observe that model selection for both 70B and 8B boosts performance. We further compare Uniform Souping with SoCE model selection with SoCE to understand the impact of weight optimization on the model soup, and observe that it leads to a relative improvement of 2.91% for 70B model and 3.36% for 8B models.

We evaluated TIES, DARE-Linear, and DARE-TIES against SoCE to quantify the impact of weight-merging on specialized capabilities. As shown in Table 2, TIES and DARE approaches suffer from catastrophic performance degradation for BFCL achieving 0% success on the Multi-Turn category of BFCL. This hints at models’ long context capability collapse post TIES/DARE merging. To validate this further, we benchmark these approaches on four other benchmarks using LM Evaluation Harness (Gao et al., 2023): Truthful Q/A (Lin et al., 2022) (0-shot MC2 Accuracy), Long-Bench (Bai et al., 2024) (0-shot average subtask score), MMLU-Pro (Wang et al., 2024) (5-shot Exact Match), and IFEval (Zhou et al., 2023) (0-shot Strict Prompt Accuracy). We present these results for 8b models in Table 3. While TruthfulQA remains stable and even gets a minor performance

Model	BFCL Accuracy
xLAM-2-70b (Prabhakar et al., 2025)	78.45%
CoALM-70B (Acikgoz et al., 2025)	54.49%
watt-tool-70B (gm8.ai, 2024a)	73.57%
functionary-medium-70B (MeetKai, 2024)	62.32%
TIES (Yadav et al., 2023)	46.47%
DARE-Linear (Yu et al., 2024)	46.33%
DARE-TIES (Yu et al., 2024)	46.24%
Uniform Souping	68.33%
Uniform Souping w/ SoCE Model Selection	78.40%
SoCE(Proposed Method)	80.68%

(a) 70 billion parameters models

Model	BFCL Accuracy
xLAM-2-8b (Prabhakar et al., 2025)	72.37%
ToolACE-2-8B (Liu et al., 2024)	68.73%
watt-tool-8B (gm8.ai, 2024b)	67.79%
BitAgent-8B (bittensor, 2024)	67.49%
CoALM-8B (Acikgoz et al., 2025)	54.52%
TIES	43.75%
DARE-Linear	44.43%
DARE-TIES	43.39%
Uniform Souping	69.80%
Uniform Souping w/ SoCE Model Selection	74.01%
SoCE(Proposed Method)	76.50%

(b) 8 billion parameter models

Table 2: BFCL Performance of 8 billion and 70 billion parameter models. The first set of entries are the ingredient models, followed by model merging methods (TIES, DARE), followed by 3 Souped Models. Uniform Souping is where all models are combined with the same weight, Uniform Souping with model selection only combines the selected models using the anti-correlation criterion and SoCE is the proposed method that augments Uniform Souping with model selection with weights optimization.

Method	d	TQA	IFEval	MMLU-Pro	Long-Bench	BFCL
ToolACE-8B	—	53.3	67.8	34.1	30.9	68.73
xLAM-8B	—	54.2	28.5	33.1	24.4	72.37
watt-8B	—	54.1	56.0	34.2	22.7	67.79
TIES	0.2	55.0	48.8	8.5	5.9	43.8
DARE-Linear	0.3	54.8	49.7	7.7	6.3	44.4
DARE-TIES	0.5	54.5	47.3	7.6	5.7	43.4
SoCE	—	54.3	64.0	38.9	24.7	76.5

Table 3: Performance of Merging Methods on 8B models: Comparison of SoCE against TIES and DARE across reasoning, instruction following, and function calling. Bold values indicate best-in-category. All metrics are reported in percentages. TIES and DARE exhibit significant regressions in MMLU-Pro and Long-Bench, whereas SoCE maintains or improves performance relative to SOTA specialized models.

boost, MMLU-Pro and LongBench scores collapse to $< 10\%$, indicating a fundamental failure in maintaining the parameter calibration required for complex reasoning and long-context processing. In contrast, SoCE achieves a superior 76.5% on BFCL and 38.9% on MMLU-Pro, outperforming both the merging baselines and the specialized models. While SoCE is not best at LongBench or TruthFul Q/A compared other models, it can be noted that its better than the best BFCL individual ingredient model (xLAM-8b) and comparable to other models. These results demonstrate that SoCE effectively synthesizes ingredient model capabilities without the reciprocal inhibition or erosion characteristic

of previous merging techniques. We observe similar results for 70 billion-sized model for TIES and DARE approaches across these benchmarks and present the same in Appendix A.2.

We present the results on the Multilingual Grade School Math (MGSM) Benchmark (Shi et al., 2022) in Table 4. Similar to BFCL, we observed that SoCE performs better than the candidate models and uniform souping. We consider 4 open source models with 6.74 billion parameters for model souping: MetaMathOctopus-7B (Chen et al., 2023), MetaMathOctopus-MAPO-DPO-7B (She et al., 2024), MathOctopus-MAPO-DPO-7B (She et al., 2024), and Mathoctopus-Parallel-7B (She et al., 2024). We present the results for checkpoints synthesized using uniform souping (Wortsman et al., 2022a), uniform souping with SoCE candidate selection, and SoCE in Table 4. Uniform souping, considering all candidates, leads to performance regression compared to the best candidate models.

We also perform similar ablations to ∞ Bench (Zhang et al., 2024) by training 5 Llama 3 architecture checkpoints (Grattafiori et al., 2024) with 70 billion parameters with variations of the same data mix to understand if souping is useful beyond tool calling and math for LLMs. We present these results in Tables 5 and 10. We can see that even though the candidate models have similar performance having been trained on

Model	MGSM Accuracy
MetaMathOctopus-7B (Chen et al., 2023)	41.9%
MetaMathOctopus-MAPO-DPO-7B (She et al., 2024)	<u>50.9%</u>
MathOctopus-MAPO-DPO-7B (She et al., 2024)	39.0%
Mathoctopus-Parallel-7B (She et al., 2024)	35.5%
Uniform Soup	47%
Uniform Soup w/ SoCE Model Selection	47.8%
SoCE (Proposed Method)	51.7%

Table 4: Results for MGSM Benchmark: First 4 rows are ingredient models and the last 3 rows are souped models. It can be seen that uniform souping of all four models leads to performance regression compared to *MetaMathOctopus-MAPO* and uniform soup with candidate selection reduces this regression. Weight tuning in SoCE improves the final performance by 1.57% relative compared to the best baseline model.

Model	∞ Bench Accuracy
Model Candidate 1	27.24%
Model Candidate 2	24.87%
Model Candidate 3	26.72%
Model Candidate 4	27.24%
Model Candidate 5	<u>27.44%</u>
Uniform Soup	27.44%
Uniform Soup w/ SoCE Model Selection	27.85%
SoCE(Proposed Method)	28.0%

Table 5: ∞ -Bench accuracy of LLAMA 3 70B architecture based models. It can be seen that model selection (corresponding to uniform souping with SoCE Model Selection row) and weight tuning (corresponding to SoCE row) leads to improvements. Overall, SoCE improves the best models candidate by 0.56%.

variants of a single data mix, model souping is still helpful in improving performance. Uniform souping does not regress performance, and uniform souping with SoCE model selection improves performance by 1.49% relatively. Further, SoCE led to a 2.04% relative (0.56% absolute) lift in performance compared to the best model candidate, also demonstrating that both weight tuning and candidate selection are important.

4.3 Large Scale Model Analysis: BFCL, MGSM, and ∞ Bench

We conducted model souping and evaluation experiments on a vast set of candidate checkpoints across the MGSM, BFCL, and ∞ Bench benchmarks to systematically investigate the effects of souping. Our analysis yielded the following key findings:

- **Increased Linear Correlation Across Cat-**

egories after Souping: Following souping, the performance metrics across different categories exhibit a marked increase in linear correlation. As illustrated in Figure 2, the top row represents the Pearson correlation coefficients of checkpoint performance prior to souping, while the bottom row represents the correlations post-souping for BFCL, MGSM, and ∞ Bench, respectively. These results indicate that souping leads to a more consistent and correlated performance profile across categories within a benchmark.

- **Consistent Performance Gains Across categories:** We observe higher average gains across the majority of categories and souping experiments. For example, in case of BFCL, the trained ~ 500 unsouped checkpoints had a mean accuracy of 49.85% (standard deviation of 4.7%) and randomly souped (~ 100) checkpoints had mean accuracy of 52.75% (standard deviation of 3.9%). We dig deeper into this aspect in section A.1.1.

The implications of these findings are twofold. First, the process of training large models is often ad hoc, with optimal performance across diverse capabilities typically achieved through extensive experimentation with model parameters and data recipes—an approach that is resource-intensive. Our results suggest that, within a given benchmark, cooperative gains can be achieved in a more deterministic and systematic manner by leveraging strong baselines and principled souping. Second, the observed overall improvements in the average metric values across souping candidates provide strong evidence of robustness of models candidates by adopting end-to-end souping methodologies.

4.4 SoCE Candidate selection experiments

SoCE comprises two components: (1) leveraging correlation patterns to identify experts for uncorrelated categories, and (2) assigning appropriate weights to these experts. We analyze the significance of checkpoint diversity and the impact of anti-correlations within benchmarks. On MGSM and BFCL, SoCE yields substantial improvements where distinct experts can be identified across anti-correlated checkpoints (Figures 4a, 4b). Where clear experts are harder to identify—such as FLORES-36 (Table 11)—or where benchmarks exhibit minimal anti-correlation, gains are more modest (Table 5, Figure 2c).

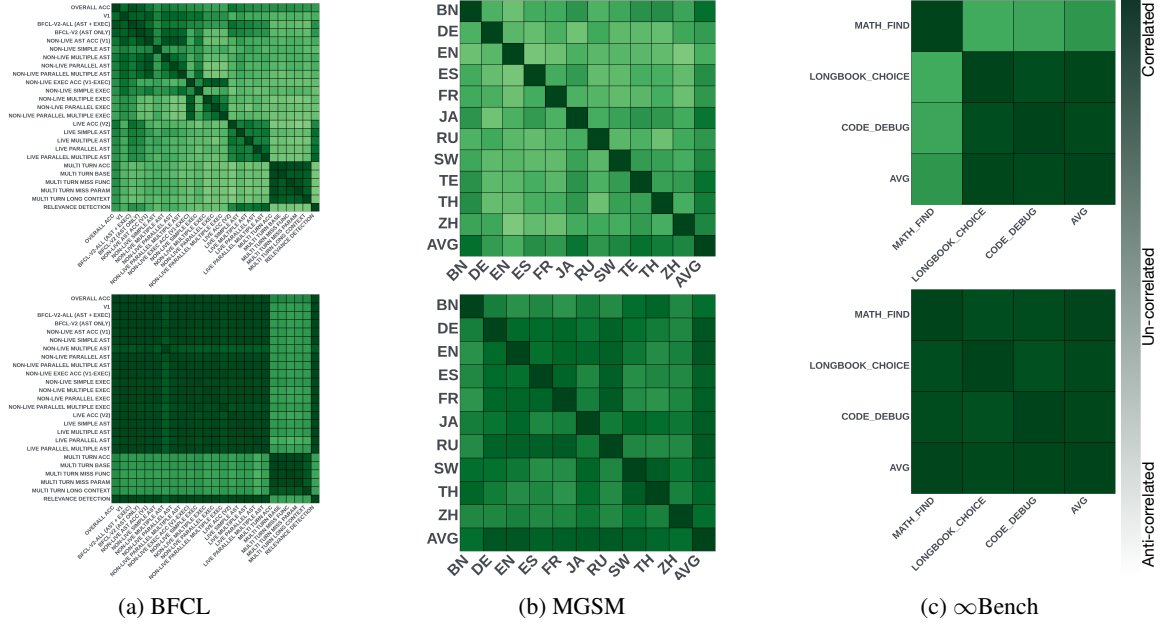


Figure 2: **Intra-benchmark performance Pearson Correlation (Pre-, and Post- souping)**: This is the pearson correlation matrix of metrics across different categories on (left to right:) 500+ unsouped (above) 100+ and randomly souped (below) checkpoints on BFCL; 40 checkpoints on MGSM unsouped (above) and souped (below), and; 80 unsouped checkpoints for ∞ Bench (above) and souped (below). We observe that after souping, the performance across all the categories become highly linearly correlated.

5 Discussion

In the general case, the SoCE can be further explored as a broader solution for adding new domain or capabilities to existing open models, overcoming overfitting (Table 9). We anticipate future research covering:

Application to many more tasks: SoCE can be widely applied to the multi-task training objectives, including

- multilingual applications: a task-specific checkpoint + a language specific checkpoint merged (like in (Akiba et al., 2025));
- application to combine capabilities at category level instead of capability level like in previous works (Cohere et al., 2025);
- enabling specific use cases, where the training data should remain private, but the checkpoint and its unique capability can be spread across the model family by souping.

Societal Impact: While the yearly increase in computational scale leads to growing inequality in academic access to frontier models (see the detailed analysis in (Kogkalidis and Chatzikyriakidis, 2024)), the presented method proposes both easily accessible low cost opportunities for the broader community efforts and also promotes the iterative reuse of the existing pretrained models, which can

lead to significant savings in computing resources. This framework alone can significantly expand the collaboration opportunities in the open source landscape: Llama-derivatives model family currently approaches 150k models¹ with groups of the same size and architecture that can be souped.

6 Conclusion

This work introduces SoCE, a category-aware, principled model souping technique that achieves state-of-the-art results and performance improvement across diverse domains. We demonstrate that this approach not only improves overall benchmark performance but also enhances consistency and robustness across categories. We hope these findings inspire further research into efficient model aggregation and reuse within the community. We believe that the presented method can propose a new perspective for the OSS community to reuse and revive derivative models of the same architecture.

Limitations

General Method Limitations

The results presented in this paper advance our understanding of model souping, but some methodological and conceptual limitations warrant more

¹<https://huggingface.co/models?search=llama>

discussion.

Benchmark structure: Our method assumes that the target benchmark ships with an explicit sub-categorical decomposition and that each sub-category contains enough data points for per-category scores to be estimated reliably across a sufficient number of models. All experiments in this paper use benchmarks with pre-defined category labels. Many benchmarks do not expose such structure, and we leave the discovery of *implicit* benchmark structure (for example, by means of clustering) to future work.

Evaluation Protocol and Test-Score Reliance: SoCE performs candidate selection and weight optimization entirely in benchmark performance space: it consumes the per-subcategory scores that public leaderboards already report for existing models, both to identify weakly correlated capability clusters and to grid-search soup weights. Our intended protocol is to derive this correlation structure and these weights from a benchmark’s *validation* split and to reserve the test split for final reporting, operating under the standard ML assumption that the two splits are drawn from the same or similar distribution. None of the benchmarks considered here provides such splits, hence we relied on the direct leaderboard scores to pick candidates. In order to understand the risk of overfitting to the BFCL benchmark given test score reliance, we evaluated the merged models on several unrelated capabilities and benchmarks and report these results in Table 9. The souped models show no significant regression while hill-climbing on BFCL, indicating that the correlation structure captured reflects genuine model properties rather than test-set artifacts. We encourage the LLM Evaluation community to release explicit validation and test splits, so that selection, tuning, and hill-climbing need not be performed directly on the test set which has been a concern that extends well beyond model merging to fine-tuning and LLM-training generally.

Application in the model training practice: In this work, we have only tested the souping of the ‘final’ post-trained and aligned checkpoints. As other works have shown, the models can be souped after pre-training (Li et al., 2025), after post-training (Wortsman et al., 2022a), with adapters². We do not recommend souping of the checkpoints from different training stages, as well as souping

unaligned or uncensored models with the aligned ones to avoid the inheritance of risks.

We also want to highlight that all the experiments are carried out on Llama 3 (Grattafiori et al., 2024) derivative models and essentially have the same pre-trained checkpoint. We are currently unaware if souping requires the same pre-trained checkpoint or can it work with different pre-trained checkpoints as well.

Scaling Laws

While model souping has demonstrated promising results, it is important to consider potential limitations and diminishing returns as more models are combined. The extent to which performance continues to improve may depend on the diversity and capabilities of the individual models being souped. We have not yet systematically tested the upper bounds of this approach, and it is possible that there exists an optimal strategy for souping checkpoints that varies according to the capability differences between the models involved. Further empirical investigation is needed to better understand these scaling dynamics.

Ethical Considerations

It is important to note that models undergoing souping procedures should be released with the disclosure and detailed description of the said procedures. Otherwise, mechanistic interpretability research on such checkpoints can be compromised: works like (Sun et al., 2024; Yu et al., 2025) would show blurred results of an unknown nature. Furthermore, souped models should be evaluated for safety and biases to make sure souping has not injected new biases or amplified existing ones.

References

- Pierre Ablin, Angelos Katharopoulos, Skyler Seto, and David Grangier. 2025. Soup-of-experts: Pretraining specialist models via parameters averaging. *arXiv preprint arXiv:2502.01804*.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Emre Can Acikgoz, Jeremiah Greer, Akul Datta, Ze Yang, William Zeng, Oussama Elachqar, Emmanouil Koukoumidis, Dilek Hakkani-Tür, and Gokhan Tur. 2025. Can a single model master both

²https://huggingface.co/docs/peft/en/developer_guides/model_merging

- multi-turn conversations and tool use? coalm: A unified conversational agentic language model. *arXiv preprint arXiv:2502.08820*.
- Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2025. [Evolutionary optimization of model merging recipes](#). *Nature Machine Intelligence*, 7(2):195–204.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multi-task benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137. Association for Computational Linguistics.
- bittensor. 2024. Bitagent-8b. <https://huggingface.co/BitAgent/BitAgent-8B>.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2023. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *arXiv preprint arXiv:2310.20246*.
- Team Cohere, :, Aakanksha, Arash Ahmadian, Marwan Ahmed, Jay Alammam, Milad Alizadeh, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, Zahara Aviv, Sammie Bae, Saurabh Baji, Alexandre Barbet, Max Bartolo, Björn Bebensee, and 211 others. 2025. [Command a: An enterprise-ready large language model](#). *Preprint*, arXiv:2504.00698.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Pradeep Dubey. 1975. [On the uniqueness of the shapley value](#). *International Journal of Game Theory*, 4:131–139.
- Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. [Linear mode connectivity and the lottery ticket hypothesis](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Lintang Sutawika, Jerome Tang, and Anish Thite. 2023. [A framework for few-shot language model evaluation](#).
- Amirata Ghorbani and James Zou. 2019. [Data shapley: Equitable valuation of data for machine learning](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR.
- gm8.ai. 2024a. Watt.ai 70b. <https://huggingface.co/watt-ai/watt-tool-70B>.
- gm8.ai. 2024b. Watt.ai 8b. <https://huggingface.co/watt-ai/watt-tool-8B>.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. [The flores-101 evaluation benchmark for low-resource and multilingual machine translation](#). *Preprint*, arXiv:2106.03193.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2023. [Editing models with task arithmetic](#). *Preprint*, arXiv:2212.04089.
- Pavel Izmailov, Dmitrii Podoprikin, T. Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. [Averaging weights leads to wider optima and better generalization](#). In *Conference on Uncertainty in Artificial Intelligence*.
- Ajay Kumar Jaiswal, Shiwei Liu, Tianlong Chen, Ying Ding, and Zhangyang Wang. 2023. Instant soup: Cheap pruning ensembles in a single pass can draw lottery tickets from large models. In *International Conference on Machine Learning*, pages 14691–14701. PMLR.

- Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. 2025. [Model stock: All we need is just a few fine-tuned models](#). *Preprint*, arXiv:2403.19522.
- Young Kyun Jang, Dat Huynh, Aditya Shah, Wei-Kang Chen, and Ser-Nam Lim. 2024. [Spherical linear interpolation and text-anchoring for zero-shot composed image retrieval](#). In *Computer Vision – ECCV 2024*, volume 15077 of *Lecture Notes in Computer Science*, pages 233–250. Springer.
- Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. [Dataless knowledge fusion by merging weights of language models](#). *ArXiv*, abs/2212.09849.
- Anat Kleiman, Gintare Karolina Dziugaite, Jonathan Frankle, Sham Kakade, and Mansheej Paul. 2025. [Soup to go: mitigating forgetting during continual learning with model averaging](#). *Preprint*, arXiv:2501.05559.
- Konstantinos Kogkalidis and Stergios Chatzikyriakidis. 2024. [On tables with numbers, with numbers](#). *Preprint*, arXiv:2408.06062.
- Yunshui Li, Yiyuan Ma, Shen Yan, Chaoyi Zhang, Jing Liu, Jianqiao Lu, Ziwen Xu, Mengzhao Chen, Minrui Wang, Shiyi Zhan, Jin Ma, Xunhao Lai, Deyi Liu, Yao Luo, Xingyan Bin, Hongbin Ren, Mingji Han, Wenhao Hao, Bairen Yi, and 7 others. 2025. [Model merging in pre-training of large language models](#). *Preprint*, arXiv:2505.12082.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252. Association for Computational Linguistics.
- Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, and 1 others. 2024. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*.
- Llama. 2025. [Llama-4-scout-17b-16e](#). Hugging Face model card. Large language model.
- Michael Matena and Colin Raffel. 2022. [Merging models with fisher-weighted averaging](#). *Preprint*, arXiv:2111.09832.
- MeetKai. 2024. Meetkai functionary medium model 70b. <https://huggingface.co/meetkai/functionary-medium-v3.1>.
- Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA. Curran Associates Inc.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. [The berkeley function calling leaderboard \(BFCL\): From tool use to agentic evaluation of large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48371–48392. PMLR.
- Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalganekar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, and 1 others. 2025. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*.
- Zixuan Qin, Qingchen Yu, Kunlin Lyu, Zhaoxin Fan, and Yifan Sun. 2026. [The Achilles’ Heel of LLMs: How altering a handful of neurons can cripple language abilities](#). *Preprint*, arXiv:2510.10238.
- Qwen. 2025. [Qwen3-30b-a3b-instruct-2507](#). Hugging Face model card. Large language model.
- Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. 2022. Diverse weight averaging for out-of-distribution generalization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. Warm: on the benefits of weight averaged reward models. *ICML’24*. JMLR.org.
- Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. 2022. [The shapley value in machine learning](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5572–5579. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, and 7 others. 2024. [Code llama: Open foundation models for code](#). *Preprint*, arXiv:2308.12950.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023.

- Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.
- Lloyd S Shapley. 1953. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton.
- Shuaijie She, Wei Zou, Shujian Huang, Wenhao Zhu, Xiang Liu, Xiang Geng, and Jiajun Chen. 2024. [Mapo: Advancing multilingual reasoning through multilingual alignment-as-preference optimization](#). *Preprint*, arXiv:2401.06838.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. [Language models are multilingual chain-of-thought reasoners](#). *Preprint*, arXiv:2210.03057.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2022. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *ArXiv*, abs/2206.04615.
- George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024. [Zipit! merging models from different tasks without training](#). *Preprint*, arXiv:2305.03053.
- Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. [Massive activations in large language models](#). *ArXiv*, abs/2402.17762.
- Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. [Challenging big-bench tasks and whether chain-of-thought can solve them](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. [Gemini: A family of highly capable multimodal models](#). *Preprint*, arXiv:2312.11805.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Joachim Utans. 1996. [Weight averaging for neural networks and local resampling schemes](#). In *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*, pages 133–138. AAAI Press.
- Jiachen T. Wang and Ruoxi Jia. 2023. [Data banzhaf: A robust data valuation framework for machine learning](#). In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6388–6421. PMLR.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: a more robust and challenging multi-task language understanding benchmark](#). In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS ’24, Red Hook, NY, USA. Curran Associates Inc.
- Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022a. [Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time](#). In *International conference on machine learning*, pages 23965–23998. PMLR.
- Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2022b. [Robust fine-tuning of zero-shot models](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. [Ties-merging: resolving interference when merging models](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.
- Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. [Adamerging: Adaptive model merging for multi-task learning](#). *ArXiv*, abs/2310.02575.
- Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. [Language models are super mario: absorbing abilities from homologous models as a free lunch](#).

In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org.

Mengxia Yu, De Wang, Qi Shan, Colorado J Reed, and Alvin Wan. 2025. [The super weight in large language models](#). *Preprint*, arXiv:2411.07191.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [Hellaswag: Can a machine really finish your sentence?](#) *Preprint*, arXiv:1905.07830.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. [\$\infty\$ Bench: Extending long context evaluation beyond 100K tokens](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

Max Zimmer, Christoph Spiegel, and Sebastian Pokutta. 2024. [Sparse model soups: A recipe for improved pruning via model averaging](#). *Preprint*, arXiv:2306.16788.

A Appendix

A.1 Additional experiments

A.1.1 FLORES-36

FLORES-101 (Goyal et al., 2021): Measures translation quality and multilingual understanding across a diverse set of language pairs. We use a subset containing translations in 18 languages to and from English. These are only used for ablation studies. We refer to this subset as FLORES-36. We provide a deep dive on FLORES-36 on the model stability and avg increase via souping. We randomly souped (in pairs) 37 checkpoints fine-tuned on Llama-70B, and for 35 out of 37 souping experiments, souped candidates had a higher metric score for more than 20 categories (out of 36) with a net positive gain observed across all categories (Figure 3). We also present an ablation on FLORES-36 in a similar setting as ∞ Bench and the results are presented in Table 11.

A.2 Comparison of SoCE with TIES and DARE approaches

In this subsection, we state the results for SoCE compared with TIES and DARE baselines. We presented the 8b results in Section 4.2. In this section, we present the same set of results for 70 billion parameter models. Table 8 shows results on 70b on these benchmarks. We observe similar regressions in performance across IFEval, MMLU-Pro and LongBench as observed for 8b models. While Truth Q/A doesn’t show gains compared to all individual ingredient models, the final DARE/TIES models are better than 2 out of 3 ingredients.

A.3 Analysis of TIES and DARE Assumptions

We conduct a deeper empirical investigation into the underlying assumptions of TIES (Yadav et al., 2023) and DARE (Yu et al., 2024). Our analysis reveals that several core assumptions fail to hold in practice, and that these failures directly explain the performance degradations observed in Tables 3, 8, and 6.

- **Near-orthogonal task vectors.** DARE assumes that task vectors derived from models sharing the same base are sufficiently aligned such that magnitude-based pruning preserves meaningful signal. However, pairwise cosine similarity between flattened task vectors averaged 0.042, indicating near-orthogonality. Pruning entries from such unaligned vectors

destroys shared signal rather than isolating task-specific components.

- **Majority sign conflicts.** TIES resolves destructive interference via majority-vote sign election. We find that 52.99% of active parameters exhibited sign conflicts (2-vs-1), forcing TIES to override minority signs for over half of all merged parameters. Furthermore, with a smaller set of candidate models, majority voting does not necessarily correlate with directional information, undermining the validity of this resolution strategy.
- **Extreme conflict among high-magnitude parameters.** TIES retains only top-magnitude parameters under the assumption that these carry consensus signal. We find the opposite: 81.63% of top-10% magnitude parameters exhibited sign conflicts, meaning the most important parameters are precisely where models disagree most.
- **Parameter droppability.** DARE assumes that a large fraction of task-vector parameters can be dropped—either randomly or by magnitude—without meaningful loss. However, indiscriminate dropping at scale may be catastrophic, as it risks disrupting critical or high-risk neurons (Qin et al., 2026).

These failures manifest clearly in the experimental results. SoCE consistently outperforms both TIES and DARE on BFCL-v3, with degradation most severe on tasks requiring long-context understanding and deep reasoning, as measured by MMLU-Pro and LongBench. We hypothesise that the sign-conflict resolution in TIES and the rescaling in DARE introduce artificial noise that disrupts the delicate, calibrated weight structure required for tasks involving deeper memory, recall, and extended context. These findings stand in contrast to prior work (Yadav et al., 2023; Yu et al., 2024) showing TIES and DARE to be superior or on par with simple model averaging. Our results demonstrate that simple averaging with principled candidate selection yields more stable and robust models, while TIES and DARE—despite preserving certain capabilities—introduce severe regressions in domains such as tool use.

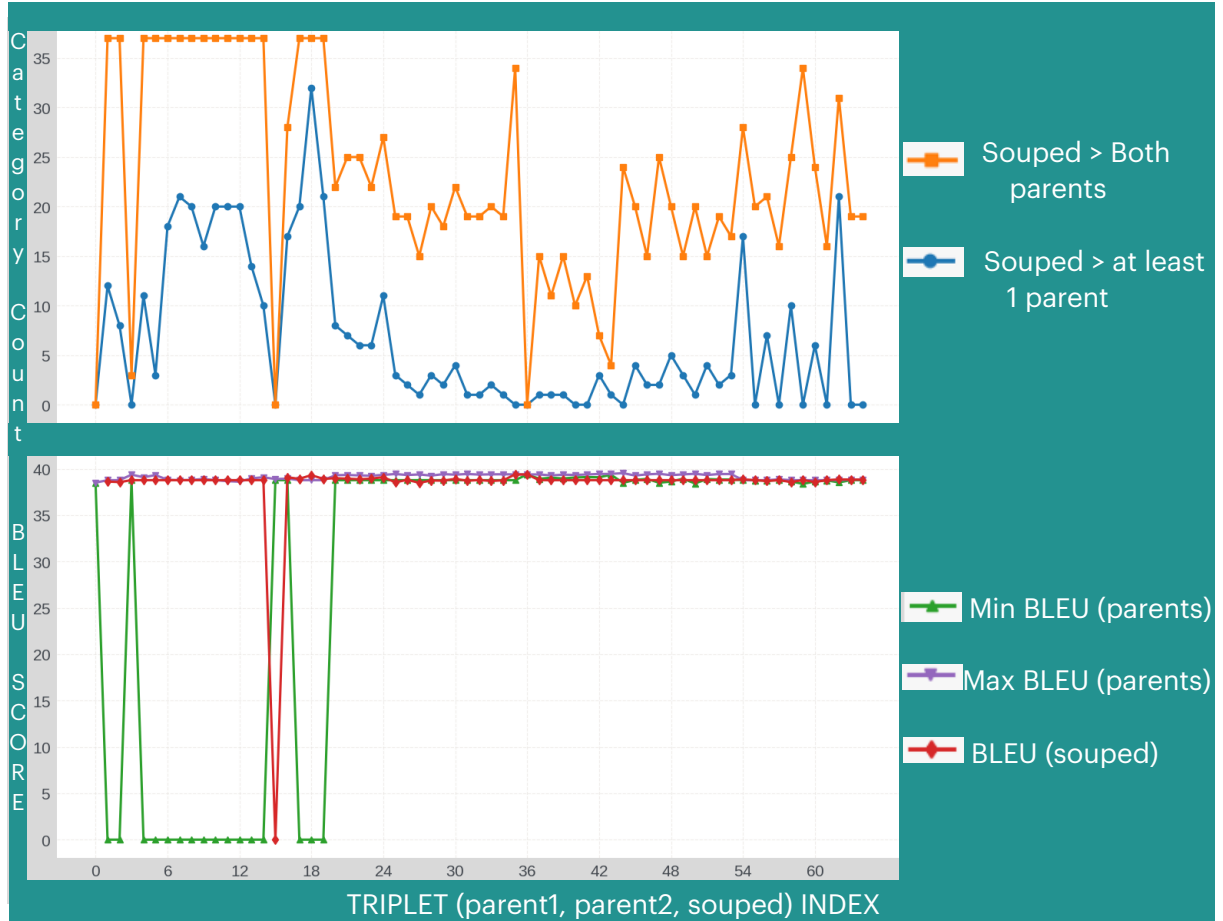


Figure 3: **Analysis of performance of 37 souped and unsouped checkpoint on Flores-36**: FLORES-36 Subset of the FLORES-101 (Goyal et al., 2021) dataset which measures translation quality and multilingual understanding across a diverse set of language pairs. The subset containing translations in 18 languages to and from English. These are only used for ablation studies. The x-axis contains the index of one souping triplet, i.e, the two parents and the souped output on the FLORES-36 benchmark. The y-axis in the top figure is the count of the number of categories in FLORES-36 and in the bottom figure, it the BLEU metric score. The orange line maps how many souped outputs have a higher score than the at least one parent per category, the blue line maps how many souped outputs have a higher score than both the souped candidates in the top figure. In the bottom figure, the green line shows the smaller average BLEU score between the parents, the purple line shows the higher BLEU score and the red line shows the souped candidate score.

Method	Density	Multi-Turn Acc	AST	Live	Relevance	Irrelevance
Baseline (ToolACE)	–	35.75%	87.77%	80.23%	72.22%	89.94%
Baseline (watt)	–	39.50%	86.67%	76.10%	77.78%	75.81%
Baseline (xLAM)	–	69.50%	84.58%	66.46%	77.78%	62.69%
DARE-Linear	0.3	0.00%	79.40%	54.20%	72.22%	74.67%
DARE-TIES	0.5	0.00%	71.56%	57.93%	44.44%	78.26%
TIES	0.2	0.00%	75.67%	55.80%	77.78%	74.42%
SoCE (Proposed)	–	70.00%	84.85%	74.32%	77.78%	77.83%

Table 6: Comparison of merging methods across BFCL evaluation metrics

A.4 Understanding Impact on Other Benchmarks while souping for one benchmark

We evaluated SoCE models that had state-of-the-art performance on BFCL-v3 on the following unrelated benchmarks to probe for overfitting to BFCL:

1. Hellaswag (Zellers et al., 2019) (commonsense reasoning via sentence completion)
2. IFEval (Zhou et al., 2023) (instruction following benchmark)
3. BIG-Bench Hard (Suzgun et al., 2022) (BIG-Bench (Srivastava et al., 2022) tasks that were hard for humans across reasoning and deduction)

We present these results in Table 9 for both 8 billion (8B) parameters model and 70 billion (70B) parameters model. We observed the SoCE models are either best (e.g. 70B model on IF-Eval) or second best (e.g. 8B on IF-Eval) in many cases and we observe no significant regressions (maximum regression of SoCE models compared to the best candidate is 1.08% for IF-Eval 8B SoCE model). This shows a strong evidence that SoCE does not lead to overfitting on the benchmark its tuned on, even with the test-score reliance as discussed in limitations section.

A.5 Deep dive on BFCL State-of-the-art Performance: Winrate Analysis of Souped Models and Individual Candidates

On the BFCL benchmark, for 70B parameters dense models, we analyze the win rate of SoCE over individual models in the soup: xLAM-2-70b-fc-r (Prabhakar et al., 2025), CoALM-70B (Acikgoz et al., 2025), and watt-tool-70B (gm8.ai, 2024a).

We first examine the proportion of tasks solved by SoCE that were also solved by each individual model. SoCE successfully completes 97.2% of the tasks solved by xLAM-2-70b-fc-r (Prabhakar et al., 2025), 97.1% of those solved by CoALM-

70B (Acikgoz et al., 2025), and 97.1% of those solved by watt-tool-70B (gm8.ai, 2024a). This indicates that SoCE retains most of the capabilities of individual models in the soup.

Next, we assess SoCE’s win rate on tasks where individual models fail. When individual models all fail on a given task, SoCE succeeds in 8.4% of cases (32 out of 380 tasks), demonstrating SoCE’s ability to solve new tasks that none of the models in the soup was able to handle.

Further, when one individual model in the soup fails on a given task, and not the others, SoCE is able to complete the task in 93.0% of cases, highlighting the robustness of the proposed souping approach.

A.6 Sensitivity Analysis of the Correlation Threshold τ and Computational Upper Bounds of SoCE

The sensitivity of SoCE to τ is bounded by two degenerate extremes: at $\tau \rightarrow -1$ the anti-correlation criterion is never satisfied and the method reduces to single-model selection; at $\tau \rightarrow 1$ it degenerates to an unconstrained soup over all B subcategory experts, at which point the grid search reaches its computational upper bound of $\binom{B+9}{9}$ candidate merges. For the benchmarks in this paper these upper bounds are 2,002 (∞ -Bench), 5,005 (BFCL), and 167,960 (MGSM). The operative regime of SoCE — small or negative τ , $N(\tau) \leq 4$, tractable grid search — lies strictly between these extremes. The anti- and de-correlation selection criterion is not merely a heuristic for complementarity; it is simultaneously the mechanism that keeps the method computationally tractable by preventing the expert count from approaching the structural upper bound B .

A.7 Post-hoc validation of our candidate selection process using Shapley Values

Data and Game Theory: Cooperative game theory, and Shapley value (Shapley, 1953), has found broad application in machine learning for quantifying component contributions. Ghorbani and Zou (2019) introduced Data Shapley, applying the Shapley value framework to equitably value individual training data points based on their marginal contribution to model performance. Rozemberczki et al. (2022) provided a comprehensive survey of Shapley value applications across

Property	SoCE (Proposed Approach)	Task Arithmetic (Ilharco et al., 2023) (TA)	RegMean (Jin et al., 2022)	Slerp + TAT (Jang et al., 2024)	CommandA (Cohe et al., 2025)
Operates on	Raw LLM weights	Task vectors	Raw weights	Embeddings	Raw LLM weights
Candidate selection	Yes (correlation-guided)	No	No	No	No (domain-designed)
Weight optimisation	Non-uniform grid search	Scalar λ , grid search	Closed-form ($X^\top X$)	Scalar α , manual	Heuristic + perturbation search
Data needed at merge	Benchmark eval scores	None	$X^\top X$ matrices	None (Slerp)	Eval scores (~500 metrics)
Modifies parameters	Yes (weighted avg)	Yes (delta ops)	Yes (reweighted avg)	No (embedding interp)	Yes (weighted avg)
Training required	No	No	No	Yes (TAT, lightweight)	Yes (purpose-built experts)
Interference handling	Complementary expert selection	None explicit	Via $X^\top X$ reweighting	N/A	Consistency regularisation
Formal guarantee	No	No	Yes (linear models only)	Yes (Slerp geometry only)	No
Scalability with N	Scales (selects subset)	Degrades	Scales (closed-form)	N/A (2-component)	Scales (7 experts; eval bottleneck)
Risk to critical neurons (Qin et al., 2026)	Low (conservative avg)	Moderate	Low-moderate	Very low (TAT only)	Low (conservative avg)
Models needed	Subset of candidate pool	≥ 2	≥ 2	1 + LoRA	All trained experts (7)
Domain	(V)LLM (Complex benchmarks)	Vision + NLP	NLP	Vision-language retrieval	LLM (enterprise, 111B)
Pretrained model used	No	As origin (implicit)	No	No	Yes (shared instruct init.)
Category structure	by within-benchmark	diverse tasks	diverse tasks	retrieval	a broad enterprise

Table 7: **Comparison of SoCE against other Model Merging methods**, showcasing its utility - operating in the parameter space, depending on publicly available evaluation protocols and selecting and souping candidates in a way that’s tractable and elegant while improving performance in the LLM space.

Method	d	TQA	IFEval	MMLU-Pro	LongBench	BFCL
xLAM-70B	—	55.7	65.3	53.6	41.53	78.45
CoALM-70B	—	51.6	67.4	41.6	36.76	54.49
watt-70B	—	60.4	73.6	61.2	43.16	73.57
TIES	0.7	57.1	58.4	7.5	2.11	46.5
DARE-Linear	0.9	59.1	66.4	6.3	2.75	46.3
DARE-TIES	0.9	57.6	61.0	8.1	2.09	46.2
SoCE	—	58.12	74.0	58.94	45.62	80.68

Table 8: Performance of Merging Methods on 70B models: Comparison of SoCE with TIES and DARE methods across Truthful QA (TQA), IFEval, MMLU-Pro and LongBench. All metrics are reported in percentages.

ML, spanning feature attribution, data valuation, model explanation, and federated learning. While alternative power indices such as the Banzhaf value have been proposed for data valuation (Wang and Jia, 2023), the Shapley value remains the canonical choice due to its uniqueness properties: Dubey (1975) proved that it is the unique allocation satisfying efficiency, symmetry, null player, and additivity-properties that make it a natural choice for analyzing model contributions in coalitional settings. Unlike prior uses of Shapley values in ML, which typically operate at the data-point or feature level, SoCE’s application is at the level of model checkpoints—a novel use case enabled by the small number of candidates (making exhaustive coalition evaluation tractable). Importantly, the Shapley value analysis in SoCE helps with post-hoc validation and interpretability rather than

optimization, confirming that the selected category experts are indeed the highest-contributing members of the coalition.

Cooperative Game Theory and the Shapley Value:

Cooperative game theory addresses situations where multiple agents collaborate as a team. A (transferable-utility) cooperative game is specified by a set of players $A = \{a_1, a_2, \dots, a_n\}$ and a characteristic function $v : 2^A \rightarrow \mathbb{R}$, which assigns to each coalition $C \subseteq A$ a real value $v(C)$ representing the utility or performance achieved by that coalition. The **Shapley value** (Shapley, 1953) is a fundamental concept in cooperative game theory, providing a principled way to quantify each player’s individual contribution to the team’s overall success, while satisfying key fairness properties (Dubey, 1975). Formally, for a player $a_i \in A$ in a game with characteristic function v , the *marginal contribution* of a_i to a coalition $C \subseteq A \setminus \{a_i\}$ is given by $m(a_i, C) = v(C \cup \{a_i\}) - v(C)$. Let Π denote the set of all permutations of the n players, i.e., each $\pi \in \Pi$ is a bijection $\pi : A \rightarrow A$. For a permutation π , define $\text{pred}(a_i, \pi)$ as the set of players preceding a_i in π . The marginal contribution of a_i in permutation π is $m(a_i, \pi) = v(\text{pred}(a_i, \pi) \cup \{a_i\}) - v(\text{pred}(a_i, \pi))$. The **Shapley value** ϕ_i for player a_i is the average marginal contribution of a_i over all possible permutations: $\phi_i =$

Models	BFCL-v3	Hellaswag	IFEval	BBH
ToolACE-2-8B	68.73%	78.92%	51.68%	44.80%
xLAM-2-8b-fc-r	72.37%	77.57%	45.00%	63.10%
watt-tool-8B (FC)	67.79%	78.59%	50.36%	36.54%
SoCE (Proposed Method)	76.50%	78.37%	50.60%	63.06%

(a) 8B Souping

Models	BFCL-v3	Hellaswag	IFEval	BBH
xLAM-2-70b	78.45%	82.80%	65.23%	82.46%
watt-tool-70B	73.57%	84.35%	73.62%	86.22%
CoALM-70B	54.49%	83.96%	67.39%	83.72%
SoCE (Proposed Method)	80.68%	83.91%	73.98%	85.72%

(b) 70B Souping

Table 9: **BFCL Souping Overfitting Results:** Performance of state-of-the-art BFCL-v3 SoCE models on Hellaswag (Zellers et al., 2019), IFEval (Zhou et al., 2023) and BIG-Bench Hard (Suzgun et al., 2022) to understand overfitting to BFCL-v3.

$\frac{1}{n!} \sum_{\pi \in \Pi} [v(\text{pred}(a_i, \pi) \cup \{a_i\}) - v(\text{pred}(a_i, \pi))]$ Equivalently, the Shapley value represents the expected increase in team value when a_i joins the team at a random position in a random ordering of the players.

Shapley Value Analysis in the Context of Model Souping:

For the Shapley value analysis, we consider a benchmark B , a metric F , and a set V of candidate models for souping. As candidates, we analyze either (1) individual checkpoints, (2) pairs, or (3) triplets, treating each candidate as a player and each subset $C \subseteq V$ as a coalition. This forms a cooperative game where the characteristic function is defined as $v : 2^V \rightarrow \mathbb{R}$, mapping each subset of candidates to the performance achieved by souping only those candidates. For any coalition $C \subseteq V$, the team performance is given by $v(C) = \text{Performance}(\text{Souped}(C); B, F)$, where $\text{Souped}(C)$ denotes the model obtained by souping the candidates in C , evaluated on benchmark B using metric F . We view the souped models as participants / agents working together to build the best model, bringing different skills / strengths to the table (as shown by the categories within the benchmarks). By souping models, we allow covering these different skills well, as showcased by an improvement in the overall metric. The Shapley value indicates the relative contributions of the sub-models using souping as the combination function in the context of the skills of the set of all models.

We define our Shapley computation by the following parameters; the benchmark is MGSM, the performance metric is average accuracy, the character-

istic function is souping and the set of candidates are (1) 4 open-source (OSS) models finetuned on Llama-7B (2) 6 pairs of the OSS models (3) 4 possible triplets these for model souping using a set of four candidate models, evaluating all possible combinations. Separately, we also apply the SoCE framework to the same set. As shown in Figure 4, our analysis reveals that model contributions are not uniform; candidates and subsets selected via SoCE exhibit significantly higher Shapley values. This finding underscores the critical role of SoCE’s candidate selection in enhancing ensemble performance. Additionally, our results indicate that disproportionately weighting a weak parent model can substantially decrease the average score (e.g., assigning weights of (0.05, 0.05, 0.9) on M4 reduces the mean to 37.0). In contrast, consistently including a strong parent model, such as M2 in combination with other candidates, reliably increases the average score. We also extend the same method of analyses to other benchmarks (see Figures 5c and 5a).

We computed Shapley values (Shapley, 1953) for a small batch of 4-5 models to understand the contributions of model candidates (single, in pairs and in triplets) and their correlation with the SoCE candidate selection mechanism. We found that SoCE’s selected candidates often contributed more significantly under the souping paradigm. We also include a scenario where category experts for uncorrelated categories are not easy to distinguish, i.e. in the case of FLORES-36. In this case, the overall improvements of souping are marginal. This also highlights the advantage of SoCE’s candidate se-

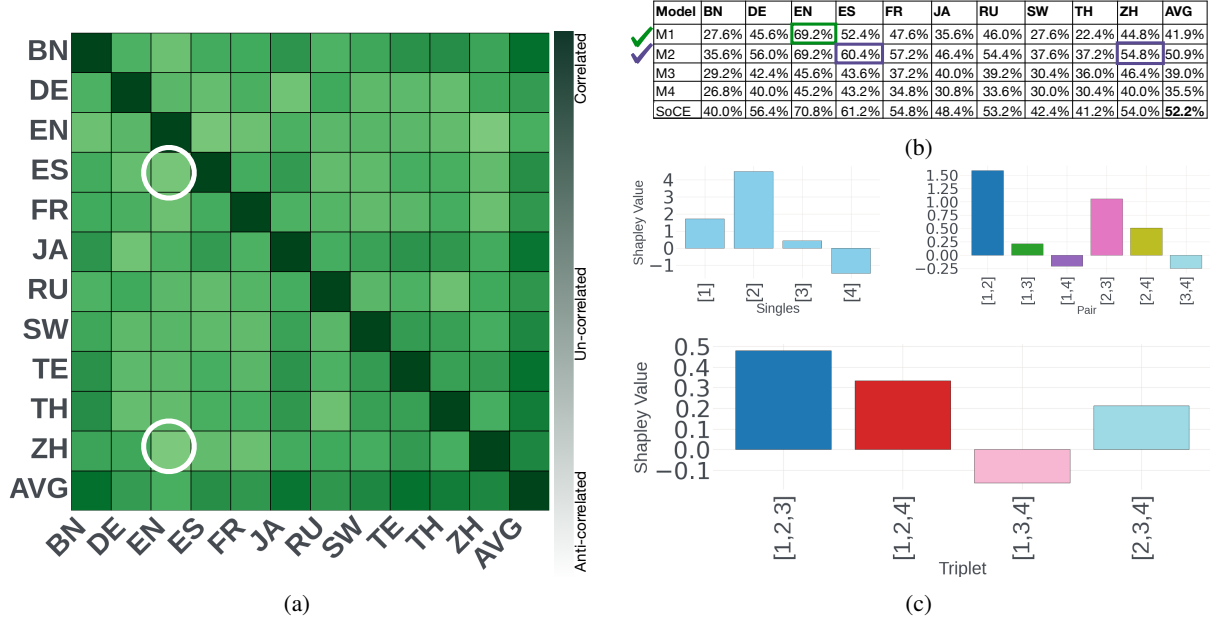


Figure 4: **Shapley value Analysis:** Figure (a) shows correlations among MGSM benchmark categories across 80 checkpoints. Table (b) reports category-wise performance for four finetuned HuggingFace models, and Figure (c) shows Shapley values for single models, pairs, and triplets. M1 and M2 emerge as experts for the least-correlated categories (ES-EN and ZH-EN) and form the strongest contributing pair. M1 is a strong parent and M4 a weak one, which the Shapley values capture—highlighting the effectiveness of SoCE’s candidate selection.

lection strategy. We present the Shapley values in Figure 5.

A.8 Qualitative examples of souping results

A.8.1 Wins when candidates fail

In tables 12 and 13, we show 2 examples of tasks where our proposed approach, SoCE, applied to 70 billion parameters dense models, solves problems that none of the individual models in the soup could solve.

For task simple_267, SoCE correctly omits the month parameter, while other models erroneously include month="upcoming".

For task java_52, SoCE correctly uses "Element.class" versus "Element" for the node_type parameter.

A.9 Souping with Large Language Models with Mixture-of-experts architecture

We conducted some baseline experiments on 6 models finetuned on Qwen/Qwen3-30B-A3B-Instruct-2507 (Qwen, 2025) and 5 models Llama-4-Scout-17B-16E (Llama, 2025) on the BFCL benchmark. We were able to find at least one souped candidate that outperformed all of the candidates. In case of checkpoints finetuned on Qwen, the category experts were not easy to distinguish, therefore, we observed only marginal improvements with

the best individual candidate’s average accuracy at 67.68% and the best souped model 68.48%. It was easier to find distinct category experts with the checkpoints finetuned with Llama4 models and we saw that the principles of SoCE could be applied. The best individual candidate’s average accuracy at 58.51% and the best souped model 61.72%. We only publish baseline results with pairwise souping as proof of concept that souping can be applied to large language models with dense as well as mixture-of-experts architectures, focusing on the former and leaving a more comprehensive study for the latter to future works.

A.10 Licenses of Open Sourced Models

In this study, we have used several Open Sourced models. We have verified that these models are released under either Creative Commons Attribution-NonCommercial or Apache 2.0 licenses.

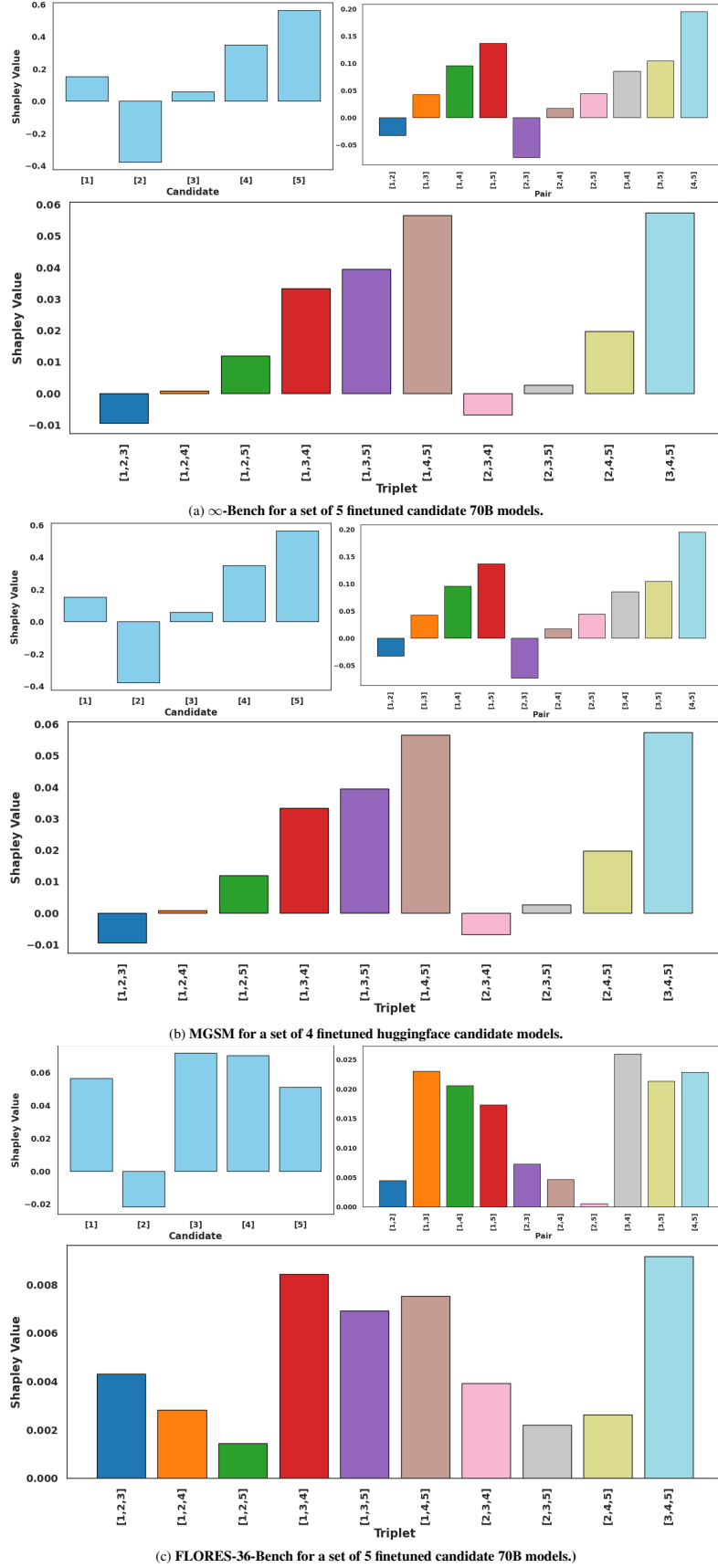


Figure 5: **Shapley values:** We computed Shapley values (Shapley, 1953) for a small batch of 4-5 models to understand the contributions of model candidates (single, in pairs and in triplets) and their correlation with the SoCE candidate selection mechanism. We found that SoCE’s selected candidates often contributed more significantly under the souping paradigm.

Model	avg_∞Bench	inf_mathfind	inf_longbookchoice	inf_codedebug
1	27.24%	33.71%	30.13%	19.80%
2	24.87%	32.57%	23.14%	19.04%
3	26.72%	34.00%	24.89%	21.32%
4	27.24%	34.00%	24.89%	22.59%
5	27.44%	32.57%	27.07%	23.10%
SoCE	27.85%	34.57%	27.51%	23.60%

Table 10: ∞Bench results (percent scale) for various models and combinations.

Model	avg_bleu	eng_deu	deu_eng	eng_spa	spa_eng	eng_tha	tha_eng	eng_por	por_eng
1	38.97	45.60	50.18	35.49	35.49	35.29	36.33	52.96	54.79
2	38.74	45.76	50.03	35.77	35.77	35.48	35.32	53.01	54.41
3	39.07	45.85	50.55	35.78	35.78	35.82	36.43	53.12	54.75
4	39.04	45.95	50.09	35.35	35.35	35.89	36.16	52.85	54.61
5	38.94	45.75	50.38	35.15	35.15	35.59	36.10	52.59	54.42
SoCE	39.19	46.22	50.39	32.06	35.72	36.26	36.57	53.27	54.83

(a) BLEU scores on FLORES-36 (German, Spanish, Thai, Portuguese.) 70B finetuned models Souping per language translation split. (Part 1 of 4)

Model	eng_vie	vie_eng	eng_ind	ind_eng	eng_hin	hin_eng	eng_fra	fra_eng	eng_ita
1	41.84	40.75	44.99	48.07	35.11	44.95	55.55	50.88	35.08
2	41.70	40.27	44.91	47.51	35.32	44.41	55.49	50.47	35.16
3	41.74	40.82	44.98	47.96	34.88	44.74	55.68	51.06	34.86
4	41.82	40.60	45.25	48.17	34.96	44.69	55.80	50.78	35.09
5	41.52	40.75	44.77	48.35	34.86	44.73	55.52	50.85	34.97
SoCE	42.08	41.06	45.22	48.50	35.47	45.08	56.15	51.26	35.26

(b) BLEU scores on FLORES-36 (Vietnamese, Indonesian, Hindi, French, Italian.) 70B finetuned models Souping per language translation split. (Part 2 of 4)

Model	ita_eng	eng_arb	arb_eng	eng_ben	ben_eng	eng_zhoHs	zhoHs_eng	eng_zhoHt	zhoHt_eng
1	38.90	34.96	45.80	29.68	37.55	33.56	34.98	17.57	31.99
2	39.20	35.32	45.59	29.94	37.10	33.32	34.43	17.39	31.22
3	38.95	35.06	46.01	29.94	37.88	33.59	34.85	18.92	31.60
4	38.52	35.43	45.97	30.13	37.92	33.58	34.41	21.28	31.53
5	38.69	35.14	46.06	29.98	37.45	33.66	34.64	18.63	31.83
SoCE	39.31	36.05	46.28	30.31	37.94	33.98	35.06	20.34	31.89

(c) BLEU scores on FLORES-36 (Italian, Arabic, Bengali, Chinese (Simplified/Traditional)) 70B finetuned models Souping per language translation split. (Part 3 of 4)

Model	eng_jpn	jpn_eng	eng_kor	kor_eng	eng_rus	rus_eng	eng_tur	tur_eng	eng_tgl	tgl_eng
1	25.20	32.83	24.44	34.67	38.63	42.32	35.78	43.03	33.37	47.80
2	24.57	32.18	24.28	33.71	38.24	41.96	36.02	42.37	33.30	47.53
3	24.97	33.64	24.66	34.93	38.57	42.22	35.78	43.44	32.38	48.29
4	24.93	33.17	23.67	34.35	38.74	42.13	35.91	42.80	32.74	48.14
5	24.75	33.34	23.87	34.53	38.77	41.88	36.26	42.93	33.00	48.17
SoCE	25.34	33.60	24.84	34.77	38.88	42.45	36.43	43.26	33.59	48.35

(d) BLEU scores on FLORES-36 (Japanese, Korean, Russian, Turkish, Tagalog.) 70B finetuned models Souping per language translation split. (Part 4)

Table 11: **Per category performance deep dive on FLORES-36 benchmark:** Each language is either translated to (language_english) or from (english_language) english. These models are less diverse, i.e, it is difficult to find category experts, and display evidence of marginal gains consistently on average and per sub-benchmark using SoCE.

Task	Description
java_52	In a Java XML processing context, help me obtain a list of all child elements of type Element from a Node representing a SAML assertion SAMLAssertionNode?
Model	Output
xLAM-2-70b-fc-r ✗	["name": "SamlObjectSignerTests.getChildren", "arguments": {"node": "SAMLAssertionNode", "node_type": "Element"}]
watt-tool-70B ✗	[SamlObjectSignerTests.getChildren(node='SAMLAssertionNode', node_type='Element')]
CoALM-70B ✗	[SamlObjectSignerTests.getChildren(node="SAMLAssertionNode", node_type="Element")]
SoCE ✓	[SamlObjectSignerTests.getChildren(node="SAMLAssertionNode", node_type="Element.class")]

Table 12: Comparison of model outputs for task java_52 in BFCL v3.

Task	Description
simple_267	Find the top rated modern sculpture exhibition happening in New York in the upcoming month
Model	Output
xLAM-2-70b-fc-r ✗	["name": "find_exhibition", "arguments": {"location": "New York City, NY", "art_form": "sculpture", "month": "upcoming", "user_ratings": "high"}]
watt-tool-70B ✗	[find_exhibition(location="New York City, NY", art_form="sculpture", month="upcoming", user_ratings="high")]
CoALM-70B ✗	[find_exhibition(location="New York", art_form="modern sculpture", month="upcoming", user_ratings="high")]
SoCE ✓	[find_exhibition(location='New York City, NY', art_form='sculpture', user_ratings='high')]

Table 13: Comparison of model outputs for task simple_267 in BFCL v3.