HIGHER-ORDER PRUNING OF EXPERTS IN MIXTURE-OF-EXPERTS LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) language models suffer from large parameter counts, which create a significant memory bottleneck. Expert pruning is the most direct approach for reducing this parameter count, yet existing methods make pruning decisions for each expert independently, and assume experts' contributions are purely additive. In reality, expert usage in MoEs is inherently cooperative. We derive HOPE (Higher-Order Pruning of Experts), a second-order pruning objective which provably minimizes an upper bound on the error resulting from pruning. We show that REAP (a state-of-the-art first-order pruning method) is a special case of HOPE where interaction terms are ignored. Across three frontier MoE models (up to 122B parameters), two distinct calibration sets, and multiple benchmarks (including math, instruction following, coding, and an agentic suite), we demonstrate that HOPE produces better pruning decisions than existing methods, and its advantage is most pronounced at high pruning rates and on challenging agentic workloads. At 50% pruning, HOPE outperforms all baselines and achieves an average rank of 1.58 out of 5 methods (versus 2.42 for the next-best method, REAP), with gains of up to +6.1% on agentic coding. Over all conditions, HOPE again achieves the best average rank and surpasses every other method in the majority of head-to-head comparisons. By preserving cooperative expert structure that first-order methods ignore, HOPE enables aggressive compression with minimal degradation, particularly on complex tasks where diverse expert combinations are invoked over long sequences.

1 INTRODUCTION

Mixture-of-expert (MoE) language models have rapidly become the dominant architecture for state-of-the-art LLMs, such as DeepSeek-V3, Qwen3.5, and GLM-4.5 (DeepSeek-AI, 2024; Qwen Team, 2025; GLM Team, 2024). MoEs attempt to scale model capacity while reducing inference costs by multiplexing the feed-forward MLP in each layer into many "expert" MLPs. Each token is independently routed to a small subset of these specialized experts in each layer, thereby reducing the inference time for each token while still allowing the model to retain high capacity and capability.

Importantly, however, although each token only activates a small subset of experts in each layer, the entire expert pool must reside in GPU memory. That is, the high parameter count of MoEs competes directly with the KV cache for valuable GPU capacity, thus limiting their practical deployment, including: 1) the need for expensive, high-memory hardware; 2) limited concurrent requests to a model; 3) reduced memory budget for context, tool calls, and reasoning.

The vast majority of the parameter count (and thus memory footprint) of MoEs is precisely the experts themselves (116 billion of the 122 billion parameters in Qwen3.5-122B-A10B are expert MLPs, taking ~ 240 GB). While quantization reduces the size of each parameter, it cannot eliminate redundant experts, and all experts still must reside in memory. *Expert pruning* takes an orthogonal approach: it permanently removes entire experts, freeing memory regardless of precision, and it composes with quantization (Lasby et al., 2025; Chen et al., 2025). Crucially, pruning is one-shot and requires no retraining. Pruning Qwen3.5-122B by 50% would immediately reduce its footprint to 125 GB, freeing 115 GB for additional context and tool calls, longer reasoning traces, and deeper searches. This substantially expands the model's effective capabilities on the same hardware.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Current expert-pruning methods have already achieved promising results, but they are limited in their independent treatment of experts. These methods are *first order*, meaning they assign each expert a scalar importance score in isolation, ignoring interactions between experts. In reality, MoE inference is inherently combinatorial: each token activates a group of K experts whose outputs are jointly aggregated, and certain pairs of experts are co-selected far more often than independent chance would predict (Figure S2). First-order methods have a major failure mode arising from this cooperative structure. If an expert A is pruned, cooperative circuits it participates in are disrupted. Partner experts that are frequently co-selected with A may no longer combine coherently, potentially degrading performance beyond what removing A alone would suggest. At aggressive pruning rates (40–50%), many such cooperative pairs are broken simultaneously, compounding damage in ways that first-order importance scores cannot predict or prevent. Higher pruning rates are also the most impactful for deployment: halving the expert pool immediately frees enough memory for longer contexts and extended reasoning, or to accommodate substantially more concurrent requests.

To this end, we derive HOPE (**H**igher **O**rder **P**runing of **E**xperts), a second-order pruning objective which considers pairwise interactions between experts to inform pruning decisions. During calibration, we record expert usage, including interaction terms that reflect how pairs of experts complement each other, resulting in a square interaction matrix per layer which is solved as a quadratic program with minimal computational overhead. We show that REAP, a state-of-the-art first-order method, is a special case of HOPE where interaction terms are ignored. Across extensive experiments (3 MoE architectures up to 122B parameters, 2 calibration sets, 6 pruning rates, and multiple benchmarks), HOPE achieved the best performance most frequently and beat every state-of-the-art baseline in the majority of head-to-head comparisons. As predicted by our theory, HOPE's margin over competing methods grows with pruning rate (particularly at 40–50%), with its advantage most pronounced in complex agentic tasks which rely on diverse expert combinations. Our analysis also confirms that HOPE's advantage stems from selectively retaining experts with stronger cooperative structure—exactly the signal that first-order methods miss.

2 Related Work

2.1 Mixture-of-experts

In a Transformer-based MoE, the MLP in each layer is replaced by E expert MLPs and a learned router. For a layer with E experts, the output is $h(x) = \sum_k g_k(x) f_k(x)$, where $f_k(x) \in \mathbb{R}^d$ is the output of expert k and $g_k(x) \in [0, 1)$ is its gate weight. The layer selects the top-K experts with the highest router logits (forming the selected set $T(x)$), and these logits are softmax-normalized into gate weights: $g_k(x) > 0$ iff $k \in T(x)$, with $\sum_{k=1}^{E} g_k(x) = 1$. This allows high total parameter count while activating far fewer parameters per token. Standard training of MoEs includes a load-balancing loss (Shazeer et al., 2017) to encourage uniform expert usage in each layer, although empirically, utilization remains highly non-uniform across experts (and this enables expert pruning).

2.2 Expert pruning

Expert pruning permanently removes experts to reduce the memory footprint of MoEs *without* retraining. Existing methods follow a common recipe: pass a calibration set $\mathcal{D}$ through the model, collect per-expert usage statistics in each layer, and assign each expert k a scalar importance score S_k. A fixed fraction of experts per layer is then pruned according to the lowest S_k. These methods differ in the statistics which are collected and how S_k is defined.

Frequency: prune the least-activated experts. Rank experts by $S_k^{\text{freq}} = |\mathcal{X}_k|$, where $\mathcal{X}_k$ is the set of tokens which activate expert k. This method is simple, but ignores expert outputs entirely.

EAN (Expert Activation Norm) (Jaiswal et al., 2025): prune by the sum of expert output norms over active tokens. Rank experts by $S_k^{\text{EAN}} = \sum_{x \in \mathcal{X}_k} \|f_k(x)\|_2$.

REAP (Router-weighted Expert Activation Pruning) (Lasby et al., 2025): prune by gate-weighted activation norm, averaged over active tokens. Rank experts by $S_k^{\text{REAP}} = \frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} g_k(x) \|f_k(x)\|_2$. This is generally considered the state-of-the-art pruning method for generative tasks, and is backed by theoretical justification.

MAN (Mean Activation Norm) (Liu et al., 2026): prune by activation norm, averaged over active tokens. Rank experts by $S_k^{\text{MAN}} = \frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} \|f_k(x)\|_2$. This is very similar to REAP, but without the gate weighting. Liu et al. (2026) also placed existing first-order pruning methods into a common framework parameterized by routing frequency, gate weighting, and activation strength.

All of these pruning methods are first-order: they score each expert independently using a scalar importance metric S_k, ignoring expert interactions.

Note that alternatives to pruning—namely expert *merging*—have also been proposed to reduce MoE parameter count, but Lasby et al. (2025) showed both theoretically and empirically that expert merging leads to irreducible error and underperforms expert pruning. Intuitively, pruning (as opposed to merging) preserves the router's ability to independently modulate surviving experts.

2.3 Expert interaction and cooperation

Non-pruning work has shown that expert co-selection patterns are stable and exploitable properties of MoEs. Flame-MoE (Kang et al., 2025) showed that expert co-selection patterns emerge early in training and stabilize throughout. Nguyen-Nhat et al. (2025) augmented the MoE router with an expert co-selection graph that encourages repeated activation of cooperative expert pairs, improving robustness. These results establish that pairs of experts *cooperate*—an exploitable property—but no prior work has leveraged this cooperative structure to inform pruning decisions.

3 HOPE: Higher-Order Pruning of Experts

In this section, we briefly outline the theoretical justification behind the HOPE objective. The full derivation can be found in Appendix A.

3.1 Setup and notation

Consider an MoE layer with E experts. For input token $x \in \mathbb{R}^d$, the layer output is $h(x) = \sum_{k=1}^{E} g_k(x) f_k(x)$, where $f_k(x) \in \mathbb{R}^d$ is the output of expert k, and $g_k(x) \in [0, 1)$ is the gate weight. The router maps x to a logit per expert and selects the K highest-logit experts to form the selected set $T(x) \subseteq \{1, \dots, E\}$, with $|T(x)| = K$. The logits of experts in $T(x)$ are softmax normalized to obtain gate weights $g_k(x)$, with $g_k(x) > 0 \Leftrightarrow k \in T(x)$ and $\sum_{k \in T(x)} g_k(x) = 1$.

Now suppose we delete a *prune-set* $P \subset \{1, \dots, E\}$ of experts from the layer. For a token x, the pruned experts in $P \cap T(x)$ (i.e. those which *would* have been selected if not removed) are replaced by the next-highest-logit experts $R(x)$, which are now in top-K ($|R(x)| = |P \cap T(x)|$). The new selected set is $(T(x) \setminus P) \cup R(x)$, with updated gate weights $g'_k(x)$, with $g'_k(x) > 0 \Leftrightarrow k \in (T(x) \setminus P) \cup R(x)$. The *pruned* layer's output is $h'(x) = \sum_{k=1}^{E} g'_k(x) f_k(x)$. Our goal is to find the prune-set P (given a prescribed budget $|P|$) that minimizes the *pruning error* $\|h(x) - h'(x)\|$.

3.2 Error decomposition

Following Lasby et al. (2025), we decompose the pruning error of a MoE layer into two components:

$$h(x) - h'(x) = \Big[\sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big] + \sum_{k \in T(x) \setminus P} (g_k(x) - g'_k(x)) f_k(x)$$

The first term is the *substitution error*, which arises from swapping the pruned experts in $P \cap T(x)$ for their replacements $R(x)$. The second term is the *renormalization error*, which arises from rescaling the gates of the retained experts.

As with Lasby et al. (2025), we note that the renormalization error is typically small (we also provide a proof of this in Appendix A), and so we focus on minimizing the substitution error.

3.3 The second-order objective

Our goal is to find the prune-set P which minimizes the substitution error:

$$P^* = \arg\min_P \Big\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big\|_2^2 \tag{1}$$

Importantly, we minimize the *squared* substitution error, which reveals interaction terms. The substitution error itself is intractable to optimize directly due to its reliance on $R(x)$ and $g'_i(x)$, which depend on the unknown prune-set P. We therefore define a tractable upper bound that depends only on pruned experts' contributions (which are observable without committing to P):

$$Z = \Big[\sum_{j \in P \cap T(x)} g_j(x) \cdot \|f_j(x)\|_2 \Big]^2 \tag{2}$$

Intuitively, Z measures the total joint contribution of the prune-set. Expanding the square reveals pairwise interaction terms between pruned experts. Note that Z depends on input tokens x and the prune-set P. This definition of Z admits the following upper bound on the substitution error:

Theorem 1. *For any pruning set P, the squared substitution error is bounded by:*

$$\Big\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big\|_2^2 \leq (1+\rho)^2 \cdot Z \quad \textit{where} \quad \rho = \frac{\max_k \|f_k(x)\|_2}{\min_k \|f_k(x)\|_2}$$

Z is a tractable surrogate whose minimization provably reduces the (squared) substitution error. The factor $(1+\rho)^2$ (independent of P) arises from bounding the contribution of replacement experts $R(x)$, whose identity depends on P and is unknown at optimization time. This factor is conservative: it assumes a worst-case replacement whose output norm equals the layer maximum. In practice, the gap between Z and the true error is far smaller than the bound suggests.

3.4 Minimizing Z

Given the upper bound in Theorem 1, our goal is to find the prune-set P which minimizes Z. More precisely, we minimize the expectation of Z over the calibration set $\mathcal{D}$: $P^* = \arg\min_P \mathbb{E}_{x \in \mathcal{D}}[Z]$. To minimize $\mathbb{E}_{x \in \mathcal{D}}[Z]$, we substitute the expanded form of Z (Equation 2) into the expectation, and replace the expectation with the empirical average. This minimization can equivalently be formulated as a binary quadratic program (QP) over the matrix $F \in \mathbb{R}^{E \times E}$, defined as follows:

$$F_{i,j} = \frac{1}{N} \sum_{x \in \mathcal{X}_{i,j}} g_i(x) g_j(x) \|f_i(x)\|_2 \|f_j(x)\|_2 \quad \text{where} \quad \mathcal{X}_{i,j} = \{x : i, j \in T(x)\} \tag{3}$$

The F-matrix is the core calibration result used by HOPE. F encodes both individual expert contributions (on the diagonal) and pairwise expert co-contributions (on the off-diagonal). High $F_{i,j}$ implies that pruning experts i, j together incurs extra error from their reinforcing contributions. To formulate the QP, we introduce binary decision variables $p_k \in \{0, 1\}$ where $p_k = 1$ iff expert $k \in P$.

Theorem 2. *Given a target pruning budget $|P|$ (the number of experts to remove per layer), the prune-set P^* which minimizes $\mathbb{E}_{x \in \mathcal{D}}[Z]$ is also the solution to the following quadratic program:*

$$P^* = \arg\min_p \; p^\top F p \quad \textit{s.t.} \quad p \in \{0,1\}^E, \sum_{k=1}^{E} p_k = |P|$$

Normalization: This formulation of F (Equation 3) minimizes $\mathbb{E}[Z]$ exactly (Theorem 2). In practice, we replace the *unconditional* average over the total number of tokens, N, with a *conditional* average, $|\mathcal{X}_{i,j}|$. This follows the analogous design choice in REAP, which prevents rarely co-selected expert pairs (which nonetheless contribute strongly) from being undervalued.

3.5 CONNECTION TO REAP

HOPE's F matrix encodes individual expert importance scores on the diagonal, and this diagonal effectively recovers the scores used by REAP to independently rank experts for pruning. Specifically, $F_{k,k} = \mathbb{E}[(g_k(x)\|f_k(x)\|)^2]$, whereas the squared REAP score is $(S_k^{\text{REAP}})^2 = \mathbb{E}[g_k(x)\|f_k(x)\|]^2$. These expressions differ only by the variance of expert contributions over activating tokens. Indeed, HOPE with zeroed off-diagonals empirically recovers the same prune-set as REAP (Figure S4). In contrast with REAP, HOPE uses the full F, where $F_{i,j}$ $(i \neq j)$ encodes the interaction between these experts when they are both activated. Thus, any improvement that HOPE achieves over REAP is directly attributable to these off-diagonal interaction terms which encode second-order effects.

3.6 PRACTICAL IMPLEMENTATION

As with other expert-pruning methods, HOPE requires a single forward pass over the calibration set. For each token, we record the active experts, their gate values, and their outputs. We also collect interaction terms, which are the product of the (gate-weighted) expert outputs when two experts are co-activated. For each layer, this gives us an $(E \times E)$ F-matrix. We solve each layer's QP (using a standard QP solver) via a continuous relaxation with the conditions in Theorem 2, rounding the continuous solution to the top $|P|$ entries. The relaxation is empirically tight: continuous solutions concentrate near 0 and 1, and the continuous–binary gap is negligible (Figure S5). In terms of computational cost, HOPE's calibration step is dominated by forward passes (shared with all methods), and solving the QP adds negligible additional cost (1–2 seconds per layer, for typical values of E).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

- **Models**: Qwen3.5-122B-A10B (256 experts, top-8 routing, 48 MoE layers), Qwen3.5-35B-A3B (256 experts, top-8 routing, 40 MoE layers), GLM-4.5-Air (128 experts, top-8 routing with 1 shared expert, 45 MoE layers); note that shared experts are never pruned
- **Baselines**: REAP (Lasby et al., 2025), EAN (Jaiswal et al., 2025), MAN (Liu et al., 2026), Frequency
- **Calibration data**: Evol-CodeAlpaca-v1 (coding-focused) (Luo et al., 2023), SWE-Bench verified trajectories (agentic-focused) (Jimenez et al., 2023)
- **Pruning rates**: 10%, 20%, 25%, 30%, 40%, 50% (prune rate of 10% means removing 10% of experts in each layer); for SWE-Bench Pro, we only test 10%, 25%, and 50%
- **Benchmarks**: Tulu3 Dev suite (GSM8K, MATH, IFEval, MMLU, BBH, TruthfulQA, PopQA, LiveCodeBench) (Lambert et al., 2024; Jain et al., 2024), SWE-Bench Pro (Deng et al., 2025)

We tested HOPE on several models ranging from 35 billion to well over 100 billion parameters, across multiple architectural families. We focused our benchmarks on agentic tasks, as well as standard coding, math, instruction following, general knowledge, etc. This gives 54 total configurations (3 models $\times$ 2 calibration sets $\times$ 6 pruning rates for Tulu, or 3 pruning rates for SWE-Bench Pro).

4.2 HOPE PERFORMANCE COMPARISON

We tested HOPE over a dense set of configurations and find that HOPE is the overall best-performing method, particularly at higher pruning rates (Figure 1). Over all conditions, HOPE had the best average rank of 2.07, finishing as the top-1 method or in the top-2 more often than any other method (top-1 in 39% of conditions, top-2 in 67% of conditions). Furthermore, in a head-to-head comparison, HOPE beat every other individual baseline in the majority of conditions, with an average head-to-head win rate of 73% (Figure S1, Tables S1–S6).

At more aggressive pruning rates (40–50%), HOPE achieved an average rank of 1.58, and finished as the top-1 method in the majority of conditions. This advantage was particularly marked in the agentic benchmark, where HOPE beat REAP in all experiments at a high pruning rate (40–50%), with a mean gap of +2.8% and up to +6.1%. This result matches our theory: HOPE's advantage over

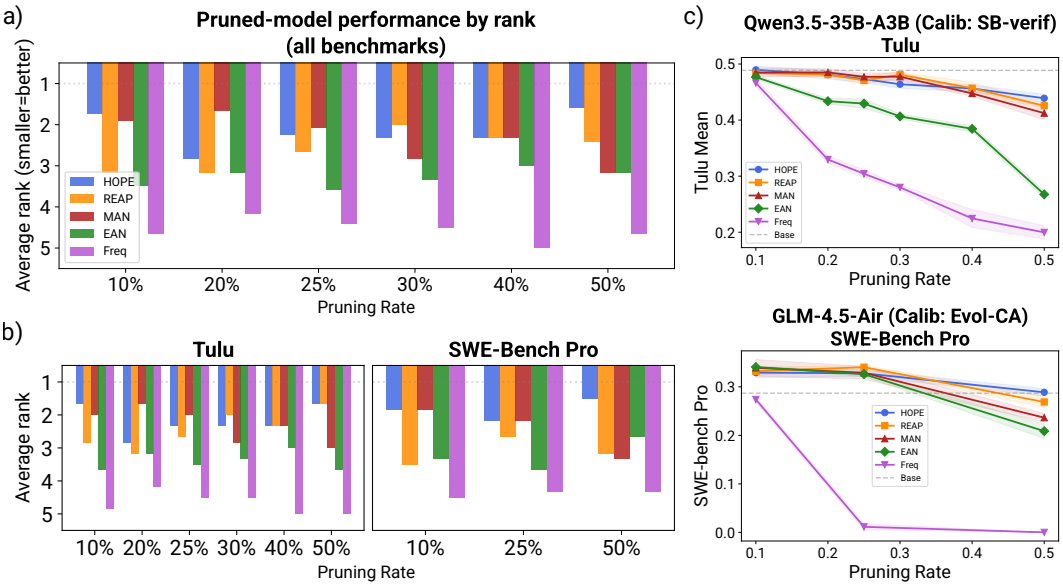


Figure 1: HOPE achieves the best overall performance across pruning conditions, and dominates at aggressive pruning rates. **a)** Average rank of HOPE compared to other baseline pruning methods, averaged over models, calibration sets, and benchmarks (54 conditions total). **b)** Average ranks separated by benchmark: Tulu3 and SWE-Bench Pro. **c)** Performance vs pruning rate for representative conditions; dashed gray line indicates the unpruned base model's performance. Note: pruned models occasionally match or slightly outperform the unpruned baseline; this is consistent with observations that calibration can act as implicit specialization, removing low-relevance experts whose contributions add noise to the target tasks (Dong et al., 2025).

first-order methods should grow with pruning rate. As more experts are removed, more interactions are broken, and the off-diagonal terms of F become more important. Higher pruning rates are also the most deployment-relevant settings, where memory savings are most crucial.

HOPE consistently improved over REAP, and this directly validates that the off-diagonal interaction terms provide signal which improves performance beyond REAP (over all conditions and pruning rates, REAP had a top-1 rate of 15% and a top-2 rate of 48%). MAN was also a strong competitor overall at lower pruning rates, but it lost its edge at more aggressive pruning rates. Furthermore, HOPE is robust: it finished last in only 1/54 conditions (20%-pruned Qwen3.5-122B-A10B), where the gap to the best method was only -1.7%. In contrast, REAP had 5 last-place finishes and Freq had 43. HOPE's worst deficit on any single condition was -4.5%, and its deficits are smaller and far less frequent, while its gains reach up to +6.1% (Figure S1, Tables S1–S6).

Interestingly, HOPE's advantage was most pronounced on agentic coding (SWE-bench Pro), code generation (LiveCodeBench), and compositional reasoning (BBH), where gains over the next-best method reached +2.0%, +19.6%, and +11.7% respectively. On simpler recall tasks (e.g. PopQA), the advantage of second-order pruning was smaller. This pattern is consistent with our theoretical motivation: tasks requiring diverse expert combinations over many tokens may benefit most from preserving cooperative structure.

4.3 Structure of HOPE's F-matrix

Having shown that HOPE outperforms first-order pruning methods—particularly at high pruning rates—we now examine the structure of the F-matrix that drives this advantage. Importantly, the off-diagonal entries of F are not noise: they exhibit noticeable substructure (Figure 2a), and their average magnitude is 33% of the diagonal (Figure 2b). Clusters of experts with high $F_{i,j}$ tend to be co-selected and contribute jointly. These interaction terms vary by layer but are consistently present throughout the network. HOPE accounts for this structure that first-order methods ignore entirely, leading to substantially different prune-sets.

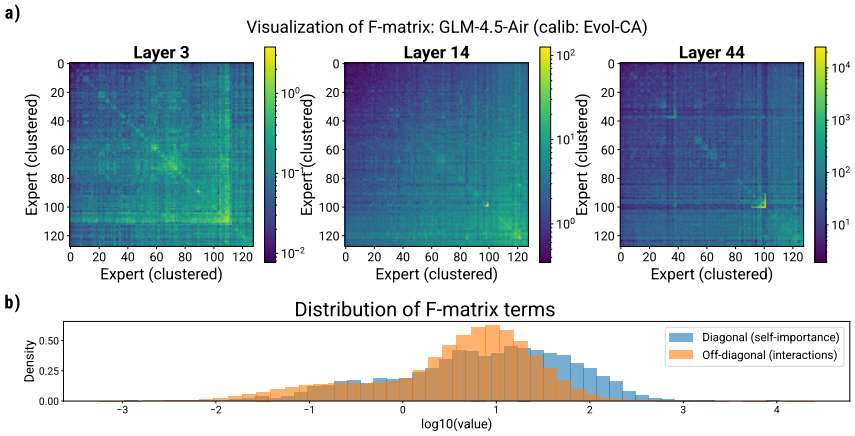


Figure 2: The interaction matrix F recorded by HOPE exhibits expert cooperation. **a)** F-matrix for GLM-4.5-Air at three representative layers (early, middle, late). Experts have been hierarchically clustered per layer for visualization. Visible block structure indicates groups of experts with high pairwise interaction scores. **b)** Distribution of diagonal vs off-diagonal terms in F across all layers (GLM-4.5-Air). Off-diagonal (interactive) terms are substantial in magnitude (compared to the diagonal), showing that pairwise interactions are a non-negligible component of the pruning objective.

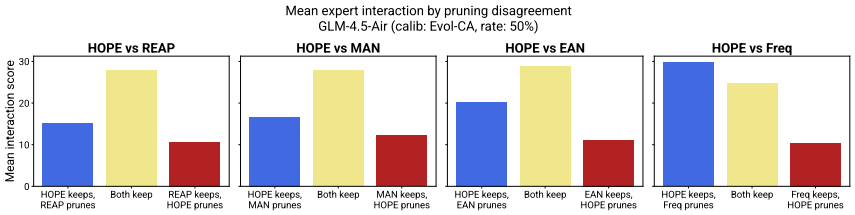


Figure 3: HOPE preferentially retains experts with stronger cooperative structure. For each baseline, we classify experts by disagreement (kept only by HOPE, kept only by the baseline, or kept by both). Experts kept by HOPE tend to have higher cooperative score (measured as the mean F score with HOPE-retained experts).

Indeed, when HOPE and a first-order method disagree, HOPE preferentially retains experts with stronger interactions with other surviving experts (measured as the mean F score) (Figure 3). Mechanistically, the QP assigns a higher cost to pruning experts which participate in cooperative clusters, causing them to be retained even if their individual importance (diagonal of F) is lower. In contrast, experts retained by a first-order method (e.g. REAP) but pruned by HOPE have high individual importance but lower interaction scores with other retained experts. As such, first-order methods tend to be misled by experts' self-importance and retain them regardless of cooperative structure.

HOPE's interaction-aware pruning decisions lead to substantially distinct prune-sets compared to other methods, but HOPE's prune-set is least distant from REAP and MAN (all three methods rely on expert outputs $f(x)$ averaged over expert-activating tokens) (Figure 4a). As expected, prune-set overlap between methods increases with pruning rate: as more experts are pruned, all methods are forced to remove the agreed-upon unimportant experts, and thereby converge on a larger shared set. At the same time, however, the remaining disagreements at higher pruning rates become more critical: the impact of each marginal pruning decision affects a larger fraction of the surviving experts, who have fewer cooperative partners. Thus, HOPE's interaction-aware decisions yield the largest gains in this high-prune-rate regime.

Disagreement between HOPE and other methods is also distributed across all layers (Figure 4b), consistent with the observation that interactions are ubiquitous throughout the network (Figure 2a, Figure S2). This indicates that second-order pruning can be valuable across the network.

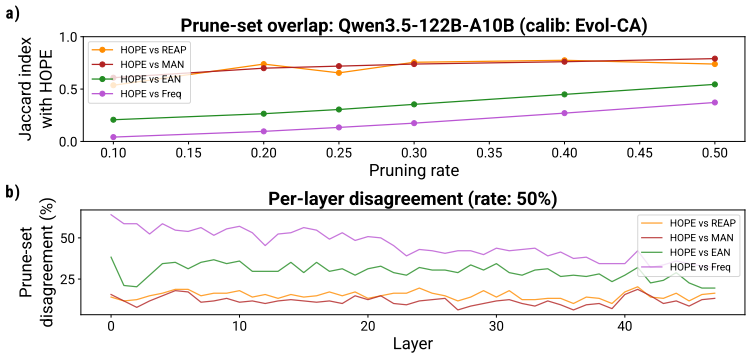


Figure 4: HOPE makes distinct pruning decisions from other baseline methods. **a)** Overlap between HOPE's pruning set versus each baseline, as a function of pruning rate (Qwen3.5-122B-A10B). We measure overlap as the Jaccard index, due to the prune-set size varying over different prune rates. **b)** For the same model and calibration set, the percentage of disagreeing pruned experts (per layer) between HOPE versus other methods at a pruning rate of 50%. Disagreement is distributed across the full network.

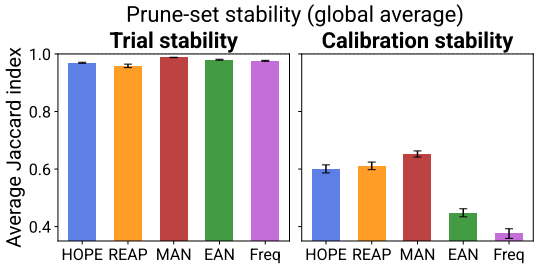


Figure 5: Prune-set stability across calibration sets and trials. **a)** Jaccard index across independent trials of the same method (averaged over models, pruning rates, and calibration sets). **b)** Jaccard index between prune-sets produced from different calibration sets: Evol-CodeAlpaca and SWE-Bench verified. Jaccard index is averaged over models, pruning rates, and trials (higher is more stability).

4.4 Calibration and Trial Robustness

Expert cooperation—as captured by HOPE's F-matrix—is a stable property of the model, not an artifact of calibration noise. Over 3 random trials, prune-set stability is high (> 0.95 Jaccard similarity on average) for all methods including HOPE (Figure 5a, Figure S3). This confirms that the interaction structure HOPE exploits is reliably recoverable from calibration data. As expected, stability across different calibration sets is somewhat lower: different calibration domains lead to slightly different pruning decisions (Figure 5b). This effect is shared across all methods, and HOPE is not more sensitive than REAP (HOPE's average calibration Jaccard index is 0.60, versus REAP's 0.61).

4.5 Compatibility with Downstream Training

Finally, we show that HOPE-pruned models serve as strong starting points for downstream training. Crucially, downstream training is not required to realize HOPE's advantage; we include this experiment to show that HOPE's structural advantage persists even after training. On REAP- or HOPE-pruned checkpoints of Qwen3.5-122B-A10B, we performed SFT on LiveCodeBench traces (collected from the unpruned model) for 1 billion tokens of training. HOPE outperformed REAP both before and after SFT (Figure 6): that is, fine-tuning narrows the gap but does not close it. This suggests that the expert structure preserved by second-order pruning provides lasting benefits that fine-tuning alone cannot replicate.

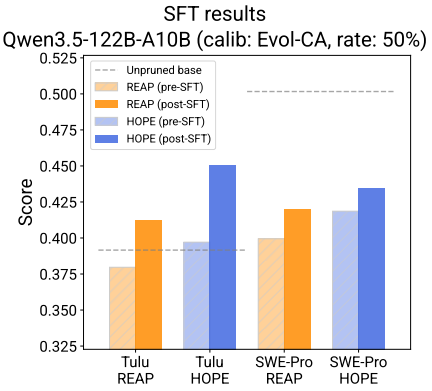


Figure 6: SFT results for pruned Qwen3.5-122B-A10B models. Pre-SFT and post-SFT use matched trials for a fair comparison. The gray dashed line indicates the performance of the unpruned model.

5 Conclusion

We presented HOPE, the first theoretically derived expert-pruning objective that directly accounts for pairwise expert interactions, formulated as a quadratic program over an interaction matrix. We showed that HOPE generalizes REAP and outperforms it (along with other methods), especially at aggressive pruning rates where interaction effects dominate. Our analysis confirms that HOPE's pruning decisions are largely influenced by off-diagonal interaction terms, which direct the selective retention of experts with high cooperative scores. Our results validate our theoretical prediction: considering these interactive terms—a signal which all first-order methods miss—in expert-pruning decisions yields better prune-sets. Importantly, HOPE's advantage over other methods is greatest in the high-pruning-rate regime, which is most relevant to real-world deployment.

Limitations: Calibration with HOPE does require additional time and memory due to its quadratic nature, although the additional cost is small. HOPE's calibration step takes about 6–7% longer than REAP on the same-size calibration set (Table S7). The additional memory cost of HOPE to store the F-matrix is also negligible for the number of experts we typically have (128–256). HOPE does require marginally more data than REAP due to its estimating inter-expert (quadratic) in addition to per-expert statistics. The difference, however, is minor: to converge to 99% agreement of the same prune-set, REAP requires 8k prompts whereas HOPE requires 10k prompts (Qwen3.5-122B-A10B, calib: Evol-CA, rate: 50%) (Figure S6). Finally, like most other methods, HOPE prunes experts in each layer independently, and does not account for cross-layer interactions.

Future work: We offer a theoretical framework (and preliminary results) for extending HOPE to cross-layer interactions and cross-layer pruning (Appendix C). Cross-layer HOPE enables non-uniform pruning budgets in each layer, potentially allocating more aggressive pruning to layers with weaker cooperative structure. In practice, however, this framework may introduce non-uniform budget-allocation instability (e.g. over-pruning layers with weak cooperative structure), which may require some additional constraints to address. Additionally, as MoEs continue to scale to larger models—where aggressive pruning is even more important for deployment—HOPE's advantage could grow even further. Finally, we note that HOPE operates only up to second-order (pairwise) interactions, which already demonstrably provide signal beyond first-order methods. Extending HOPE to even higher-order interactions could be valuable, although they are combinatorially difficult to quantify.

AI USE STATEMENT

In this work, we used generative AI to:

- Search existing literature for relevant directions and baselines
- Check human-written code for bugs
- Write the first draft of code specifically for creating plots/figures
- Suggest rewordings for specific sentences in the manuscript
- Insert citations into the manuscript and download *.bib citations
- Reformat CSV files of collected data/results into LaTeX-formatted tables (via an AI-generated script)

We did *not* use AI to write any code which created any main results, other than to identify bugs (which were subsequently verified and fixed by a human). AI was used to write the first draft of code for creating plots and figures, and this code was then verified and revised by a human.

We did *not* use AI to generate research ideas, derive theoretical proofs, design experiments, or interpret results.

We did *not* use AI to write any part of the text for the manuscript, other than as a "copy-editor" to search for typos, insert citations, and suggest rewordings for specific sentences or sections when prompted. AI was used to automate the formatting of LaTeX tables from saved CSV files.

We take responsibility for the final content of this work.

ETHICS STATEMENT

This research focuses on algorithmic compression of MoE language models. All datasets (calibration or evaluation) are publicly available and do not contain personal information. All models used are also publicly available and open source. Our work's contribution is methodological: we derived and demonstrated a method for reducing the memory footprint of MoEs. Although we hope our work will improve accessibility of generative AI, we also acknowledge that such algorithms could potentially lower barriers to deploying models/agents for harmful uses, as well. Additionally, we pruned pre-trained models, and any biases already present in those models may persist after pruning. In general, we believe our work does not introduce any new ethical considerations beyond those already present in the efficient deployment of LLMs.

REPRODUCIBILITY STATEMENT

Code for HOPE calibration, QP solving, and all analyses will be released upon publication. The theoretical derivation of HOPE is provided in full in Appendix A. Section 4.1 details the experimental setup, including models, calibration datasets, pruning rates, and evaluation benchmarks. Appendix D provides supplementary methods describing all analysis procedures in sufficient detail for reproduction, including hyperparameters, solver settings, and evaluation configurations. All experiments use publicly available models and datasets.

REFERENCES

Yixiao Chen, Yanyue Xie, Ruining Yang, Wei Jiang, Wei Wang, Yong He, Yue Chen, Pu Zhao, and Yanzhi Wang. Collaborative compression for large-scale moe deployment on edge. *arXiv preprint arXiv:2509.25689*, 2025.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Vijay Bharadwaj, Jeff Holm, Raja Aluri, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 2025.

Zican Dong, Han Peng, Peiyu Liu, Wayne Xin Zhao, Dong Wu, Feng Xiao, and Zhifeng Wang. Domain-specific pruning of large mixture-of-experts models with few-shot demonstrations. *arXiv preprint arXiv:2504.06792*, 2025.

GLM Team. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Ajay Jaiswal, Junda Wang, Ying Li, Ping Li, Tianlong Chen, Zhangyang Wang, Cong Wang, and Xinya Du. Finding fantastic experts in MoEs: A unified study for expert dropping strategies and observations. *arXiv preprint arXiv:2504.05586*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Hao Kang, Zichun Yu, and Chenyan Xiong. Flame-moe: A transparent end-to-end research platform for mixture-of-experts language models. *arXiv preprint arXiv:2505.20225*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. *arXiv preprint arXiv:2510.13999*, 2025.

Zongfang Liu, Jinghui Zhang, Zijian Ma, Guangyi Chen, and Xin Yuan. How to score experts for one-shot moe expert pruning: A unified formulation and selection principle. *arXiv preprint arXiv:2606.15716*, 2026.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023.

Minh-Khoi Nguyen-Nhat, Rachel S. Y. Teo, Laziz Abdullaev, Maurice Mok, Viet-Hoang Tran, and Tan Minh Nguyen. Modeling expert interactions in sparse mixture of experts via graph structures. *arXiv preprint arXiv:2510.16411*, 2025.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

A FULL DERIVATION OF HOPE

A.1 SETUP

For an arbitrary MoE layer with E experts, the output of that layer on input-token representation x is:

$$h(x) = \sum_{k=1}^{E} g_k(x) f_k(x)$$

where $g_k(x) \in [0, 1)$ is the softmax-normalized gate weight and $f_k(x) \in \mathbb{R}^d$ is the expert output. We have $g_k(x) > 0$ iff $k \in T(x)$ (the top-K selected set), and $\sum_{k=1}^{E} g_k(x) = 1$.

Now suppose we prune a set of experts P. For any token x, the pruned experts $P \cap T(x)$ are replaced by the next-highest-logit experts $R(x)$, with $|R(x)| = |P \cap T(x)|$. After pruning, the new gate weights are $g'_k(x) > 0$ iff $k \in (T(x) \setminus P) \cup R(x)$, and the new layer output is:

$$h'(x) = \sum_{k=1}^{E} g'_k(x) f_k(x)$$

A.2 ERROR DECOMPOSITION

The pruning error decomposes as:

$$h(x) - h'(x) = \underbrace{\Big[\sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big]}_{\text{substitution error}} + \underbrace{\sum_{k \in T(x) \setminus P} (g_k(x) - g'_k(x)) f_k(x)}_{\text{renormalization error}}$$

The first term is the *substitution error* and the second is the *renormalization error*. The substitution error is the direct effect of swapping experts, and is the dominant source of error. The renormalization error is the indirect effect of pruning on untouched experts. Because gate values $g_k(x)$ must sum to 1, pruning and replacing experts causes the gate values of non-pruned experts in the top-K set ($T(x) \setminus P$) to be rescaled. This particular decomposition was described in (Lasby et al., 2025). From this point forward, the derivations will diverge substantially.

A.3 GATE-MASS LEMMA

An intermediate result which will be helpful is to show that replacement experts in $R(x)$ have a lower total gate mass. More formally:

Lemma 3 (Replacement experts have lower gate mass)**.** *Under top-K routing with softmax normalization, suppose experts $P \cap T(x)$ are pruned and replaced with experts $R(x)$. Then* $\sum_{i \in R(x)} g'_i(x) \leq \sum_{j \in P \cap T(x)} g_j(x)$.

Proof: Define $l_k(x)$ be the router logit for expert k. Note this is a function of the input token and the router, and is unchanged by pruning.

Recall, after pruning, the new selected set becomes $T'(x) = (T(x) \setminus P) \cup R(x)$. We define the following quantities:

- $S_P = \sum_{j \in P \cap T(x)} \exp(l_j(x))$ (sum of exponentiated logits for pruned experts)
- $S_T = \sum_{k \in T(x) \setminus P} \exp(l_k(x))$ (sum of exponentiated logits for retained experts)
- $S_R = \sum_{i \in R(x)} \exp(l_i(x))$ (sum of exponentiated logits for replacement experts)

Before pruning, the gate weights are a softmax over $T(x)$. Then the total gate mass of the pruned experts (before pruning) is $\sum_{j \in P \cap T(x)} g_j(x) = \frac{S_P}{S_T + S_P}$.

After pruning, the gate weights are a softmax over $T'(x)$. Then the total gate mass of the replacement experts (after pruning) is $\sum_{i \in R(x)} g'_i(x) = \frac{S_R}{S_T + S_R}$.

Crucially, the logit $l_j(x)$ for any pruned expert $j \in P \cap T(x)$ must be *at least* the logit $l_i(x)$ for any replacement expert $i \in R(x)$. This is because every pruned expert $j \in P \cap T(x)$ had a top-K logit, and every replacement expert did not have a top-K logit. Since exponentiation is a monotonically increasing function, we have that $S_P \geq S_R$.

To complete the proof for Lemma 3, we note that we simply need to show $\frac{S_P}{S_T + S_P} \geq \frac{S_R}{S_T + S_R}$. Since all quantities are strictly positive, we can cross-multiply and see that this is true precisely when $S_P \geq S_R$ (which we showed just above).

A.4 Renormalization error is small

Recall, the pruning error (which we wish to minimize) decomposes into the substitution error and renormalization error. Next, let us show that the renormalization error is generally small.

For simplicity, we assume the new gate values are renormalized directly from the old ones (without retaking the softmax):

$$g'_k(x) = \frac{g_k(x)}{1 - \sum_{j \in P \cap T(x)} g_j(x) + \sum_{i \in R(x)} g'_i(x)}$$

Substituting into the renormalization error:

$$\sum_{k \in T(x) \setminus P} (g_k(x) - g'_k(x)) f_k(x) = \frac{\sum_{i \in R(x)} g'_i(x) - \sum_{j \in P \cap T(x)} g_j(x)}{\sum_{i \in R(x)} g'_i(x) - \sum_{j \in P \cap T(x)} g_j(x) + 1} \sum_{k \in T(x) \setminus P} g_k(x) f_k(x)$$

Note that the scalar factor in front is small when $\sum_{j \in P \cap T(x)} g_j(x) \approx \sum_{i \in R(x)} g'_i(x)$. Generally, this is expected because we typically prune experts with the lower gate probabilities among the top-K, and replace them with other experts whose gate probabilities barely missed the cut-off for top-K. We therefore focus on minimizing the substitution error.

A.5 Expanding the squared substitution error

We focus on minimizing the *squared* substitution error, as this reveals interaction terms between pruned experts. We decompose the quadratic form into three components:

$$\begin{aligned}
&\left\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \right\|_2^2 \\
&= \underbrace{\left\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) \right\|_2^2}_{A} \underbrace{- 2 \left\langle \sum_{j \in P \cap T(x)} g_j(x) f_j(x), \sum_{i \in R(x)} g'_i(x) f_i(x) \right\rangle}_{B} + \underbrace{\left\| \sum_{i \in R(x)} g'_i(x) f_i(x) \right\|_2^2}_{C}
\end{aligned}$$

For ease of notation, we split the substitution error into three components: $A + B + C$.

Intuitively, A is the contribution of the pruned experts. Note A is quadratic, and the cross-terms of A captures how much pruned experts reinforce each other. If two pruned experts contribute cooperatively, their joint removal incurs a penalty which is worse than their individual removals.

C is the contribution of the replacement experts, and it measures the contribution of these experts to the output. C also contributes to error because the replacements are not perfectly aligned with what was pruned.

Finally, B is the alignment between pruned experts and retained experts. If replacement experts are well-aligned with the pruned experts, B is large in magnitude and contributes a large negative to the substitution error.

A.6 Upper-bounding the substitution error via Z

By the Cauchy–Schwarz inequality, we have that $B \leq |B| \leq 2\sqrt{A}\sqrt{C}$. Substituting, we have the following inequality:

$$\Big\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big\|_2^2 = A + B + C \leq A + 2\sqrt{A}\sqrt{C} + C = \big(\sqrt{A} + \sqrt{C}\big)^2$$

We now upper-bound both $\sqrt{A}$ and $\sqrt{C}$ in terms of a common quantity. Define:

$$Z = \Big(\sum_{j \in P \cap T(x)} g_j(x) \|f_j(x)\|_2 \Big)^2$$

Bounding $\sqrt{A}$:

By the triangle inequality:

$$\sqrt{A} = \Big\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) \Big\|_2 \leq \sum_{j \in P \cap T(x)} g_j(x) \|f_j(x)\|_2 = \sqrt{Z}$$

Bounding $\sqrt{C}$:

Again by the triangle inequality:

$$\sqrt{C} = \Big\| \sum_{i \in R(x)} g'_i(x) f_i(x) \Big\|_2 \leq \sum_{i \in R(x)} g'_i(x) \|f_i(x)\|_2$$

We have $\sum_{i \in R(x)} g'_i(x) \|f_i(x)\|_2 \leq \max_k \|f_k(x)\|_2 \sum_{i \in R(x)} g'_i(x)$, by replacing the replacement experts' output norms with the maximum norm over all experts in the layer.

By Lemma 3, $\max_k \|f_k(x)\|_2 \sum_{i \in R(x)} g'_i(x) \leq \max_k \|f_k(x)\|_2 \sum_{j \in P \cap T(x)} g_j(x)$.

Furthermore, for each pruned expert j, we also have that $g_j(x) \leq g_j(x) \frac{\|f_j(x)\|_2}{\min_k \|f_k(x)\|_2}$. Substituting in for $g_j(x)$, we have:

$$\begin{aligned} \max_k \|f_k(x)\|_2 \sum_{j \in P \cap T(x)} g_j(x) &\leq \max_k \|f_k(x)\|_2 \sum_{j \in P \cap T(x)} g_j(x) \frac{\|f_j(x)\|_2}{\min_k \|f_k(x)\|_2} \\ &= \frac{\max_k \|f_k(x)\|_2}{\min_k \|f_k(x)\|_2} \sqrt{Z} \end{aligned}$$

This substitution provides a loose (but strict) upper bound, without relying on knowledge of the replacement experts, which makes it tractable to optimize. Thus, $\sqrt{C} \leq \rho\sqrt{Z}$, where $\rho = \frac{\max_k \|f_k(x)\|_2}{\min_k \|f_k(x)\|_2}$.

Together :

$$\Big\| \sum_{j \in P \cap T(x)} g_j(x) f_j(x) - \sum_{i \in R(x)} g'_i(x) f_i(x) \Big\|_2^2 \leq (\sqrt{A} + \sqrt{C})^2 \leq (\sqrt{Z} + \rho\sqrt{Z})^2 = (1+\rho)^2 \cdot Z$$

This completes the proof of Theorem 1.

A.7 MINIMIZING Z AS A QUADRATIC PROGRAM

Our goal is now to minimize $\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[Z]$ over the calibration set $\mathcal{D}$. We introduce binary decision variables $p_k \in \{0,1\}$, where $p_k = 1$ iff expert $k \in P$, and indicator variables $\mathbf{1}[k \in T(x)]$ which denote if expert k is in the top-K.

Substituting:

$$\begin{aligned} Z &= \big(\sum_{j=1}^{E} p_j \cdot \mathbf{1}[j \in T(x)] \cdot g_j(x)\|f_j(x)\|_2\big)^2 \\ &= \sum_{j=1}^{E}\sum_{k=1}^{E} p_j p_k \cdot \mathbf{1}[j \in T(x)] \cdot \mathbf{1}[k \in T(x)] \cdot g_j(x)g_k(x)\|f_j(x)\|_2\|f_k(x)\|_2. \end{aligned}$$

Taking the empirical expectation over the calibration set (with N tokens):

$$\begin{aligned} \mathop{\mathbb{E}}_{x}[Z] &= \frac{1}{N}\sum_{x}\sum_{j=1}^{E}\sum_{k=1}^{E} p_j p_k \cdot \mathbf{1}[j \in T(x)] \cdot \mathbf{1}[k \in T(x)] \cdot g_j(x)g_k(x)\|f_j(x)\|_2\|f_k(x)\|_2 \\ &= \sum_{j=1}^{E}\sum_{k=1}^{E} p_j p_k \underbrace{\frac{1}{N}\sum_{x\in\mathcal{X}_{j,k}} g_j(x)g_k(x)\|f_j(x)\|_2\|f_k(x)\|_2}_{F_{j,k}} \\ &= p^\top F p \end{aligned}$$

where we define:

$$F_{i,j} = \frac{1}{N}\sum_{x\in\mathcal{X}_{i,j}} g_i(x)g_j(x)\|f_i(x)\|_2\|f_j(x)\|_2, \quad \mathcal{X}_{i,j} = \{x : i,j \in T(x)\}$$

and p is a binary vector of size E.

Note that in practice, we use *conditional* normalization in F instead of unconditional normalization. This equates to the following definition of F:

$$F_{i,j} = \frac{1}{|\mathcal{X}_{i,j}|}\sum_{x\in\mathcal{X}_{i,j}} g_i(x)g_j(x)\|f_i(x)\|_2\|f_j(x)\|_2$$

This is the empirical choice which we found helps decouple interaction strength from co-selection frequency. This helps ensure that specialist expert pairs which are rarely co-selected—but contribute strongly—are not undervalued, consistent with the analogous choice made by REAP.

Note that if an expert pair is never activated ($\mathcal{X}_{i,j} = \emptyset$), then $F_{i,j} = 0$.

Therefore, finding the optimal prune-set of size $|P|$ reduces to the binary quadratic program:

$$P^* = \mathop{\arg\min}_{p\in\{0,1\}^E,\ \sum_k p_k = |P|} p^\top F p$$

This completes the proof of Theorem 2.

B SUPPLEMENTARY FIGURES AND TABLES

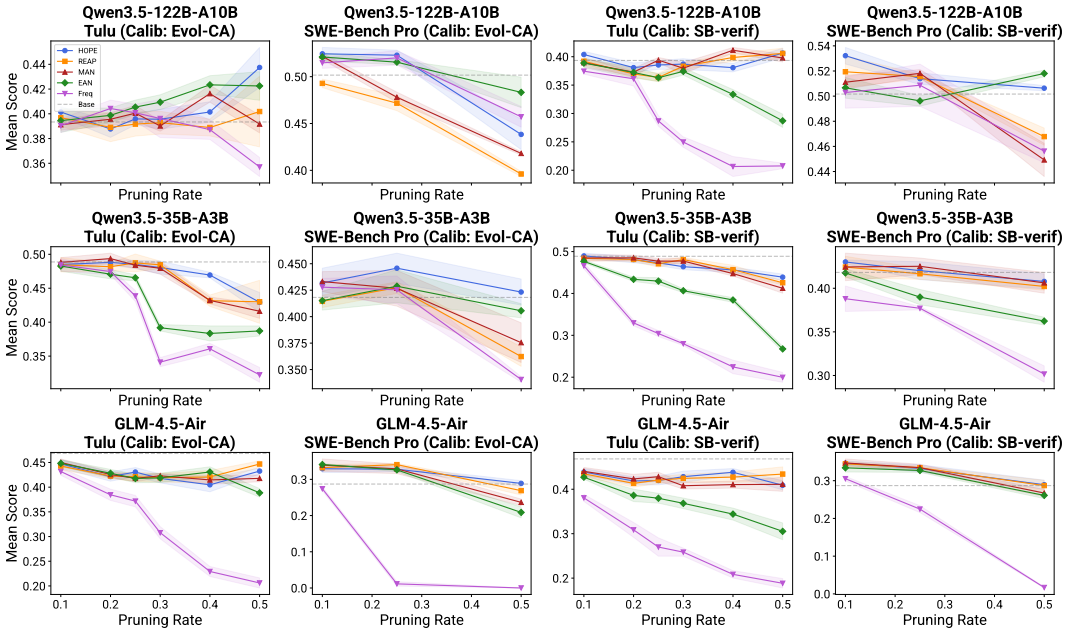


Figure S1: Full performance results across all models and calibration sets. Each row corresponds to a model architecture (Qwen3.5-122B, Qwen3.5-35B, GLM-4.5-Air). Columns alternate between mean Tulu score and SWE-Bench Pro score for each calibration set (Evol-CodeAlpaca and SWE-Bench verified). The dashed gray line indicates the performance of the unpruned base model. HOPE (blue) is competitive or best at all pruning rates, with the cleanest separation at aggressive pruning rates.

Calib	Method	Rate	GSM8K	MATH	IFEval	PopQA	MMLU	BBH	TQA	LCB	Mean
	Base		92.17±0.36	41.54±0.37	21.21±1.71	0.35±0.00	65.15±0.31	40.34±0.88	65.92±0.26	64.30±0.61	48.87±0.56
Evol-CA	HOPE	10%	94.44±0.36	42.61±0.83	19.39±0.86	**0.35±0.00**	**63.87±0.25**	**41.51±0.83**	63.13±0.41	62.63±0.66	48.49±0.53
	REAP	10%	93.43±1.89	42.37±1.15	**20.00±1.48**	**0.35±0.00**	62.74±0.94	39.37±0.36	64.53±0.81	64.84±1.84	48.45±1.06
	MAN	10%	**95.20±0.71**	**42.71±0.94**	18.79±0.86	**0.35±0.00**	63.86±0.45	**41.51±1.65**	**64.64±0.12**	63.44±0.88	**48.81±0.70**
	EAN	10%	94.44±0.36	42.13±1.47	18.79±0.86	**0.35±0.00**	62.27±0.81	39.88±1.44	64.33±0.48	63.98±0.73	48.27±0.77
	Freq	10%	93.18±0.62	42.64±2.04	19.39±1.71	**0.35±0.00**	62.94±0.21	39.47±0.38	63.39±0.90	**66.13±1.52**	48.44±0.92
	HOPE	20%	**94.70±0.62**	42.96±0.59	18.18±0.00	**0.35±0.00**	68.01±0.86	39.62±0.69	63.71±0.16	62.96±0.53	48.81±0.43
	REAP	20%	92.42±0.00	42.13±1.36	18.79±0.86	**0.35±0.00**	64.70±0.92	40.24±0.83	**63.87±0.39**	62.85±0.40	48.17±0.59
	MAN	20%	92.93±0.71	**43.13±1.04**	**20.61±0.86**	**0.35±0.00**	**68.51±0.71**	**41.62±1.71**	63.26±0.15	64.52±1.21	**49.36±0.80**
	EAN	20%	92.42±1.24	41.81±0.13	18.79±1.71	**0.35±0.00**	59.84±0.92	37.42±1.11	61.32±0.04	64.57±0.27	47.07±0.68
	Freq	20%	93.18±0.62	40.10±0.40	18.79±0.86	**0.35±0.00**	59.95±0.29	38.75±0.64	62.85±0.21	**65.65±0.80**	47.45±0.48
	HOPE	25%	93.69±0.36	40.34±1.73	18.18±0.00	**0.35±0.00**	64.50±0.70	**40.59±0.72**	64.74±0.15	**65.70±1.19**	48.51±0.61
	REAP	25%	93.69±0.71	**43.84±1.70**	18.79±0.86	**0.35±0.00**	**66.12±4.40**	38.39±1.33	64.20±1.16	64.46±0.33	**48.73±1.31**
	MAN	25%	**94.19±0.71**	43.47±0.54	18.79±0.86	**0.35±0.00**	63.16±0.28	38.80±0.38	**64.78±0.21**	63.66±0.73	48.40±0.46
	EAN	25%	91.41±0.71	41.52±0.39	**21.82±0.00**	**0.35±0.00**	58.35±0.77	38.04±1.09	59.42±0.27	61.40±0.40	46.54±0.45
	Freq	25%	91.92±1.89	37.82±1.07	17.58±1.71	**0.35±0.00**	48.53±0.34	33.64±0.69	59.33±0.34	61.72±0.33	43.86±0.80
	HOPE	30%	**95.20±0.71**	40.34±1.62	18.18±1.48	**0.35±0.00**	**66.27±1.42**	36.15±1.35	**65.26±0.03**	62.63±1.07	48.05±0.96
	REAP	30%	94.44±0.71	**42.74±1.24**	18.18±0.00	**0.35±0.00**	64.75±0.39	**39.01±0.69**	62.79±0.27	**65.32±1.08**	**48.45±0.55**
	MAN	30%	94.44±0.36	41.07±0.93	**19.39±1.71**	**0.35±0.00**	64.69±1.06	38.75±1.55	64.05±0.41	60.81±0.95	47.94±0.87
	EAN	30%	92.42±0.00	39.09±1.03	16.36±0.00	**0.35±0.00**	42.74±0.54	35.43±0.25	56.33±0.56	30.70±1.41	39.18±0.47
	Freq	30%	91.16±0.36	21.57±0.64	15.76±0.86	**0.35±0.00**	35.00±1.17	33.49±0.51	54.32±0.60	21.24±1.22	34.11±0.67
	HOPE	40%	93.69±0.36	**40.27±0.48**	18.18±0.00	**0.35±0.00**	**60.03±0.31**	36.20±0.45	**64.40±0.51**	**62.42±1.15**	**46.94±0.41**
	REAP	40%	93.18±0.62	39.43±0.96	18.18±1.48	**0.35±0.00**	52.39±1.16	37.01±1.02	62.82±0.41	41.94±1.05	43.16±0.84
	MAN	40%	**93.94±0.62**	36.90±1.06	**19.39±0.86**	**0.35±0.00**	49.86±0.10	**38.09±0.07**	64.31±0.26	42.80±1.12	43.20±0.51
	EAN	40%	92.17±0.94	39.29±0.63	14.55±0.00	**0.35±0.00**	32.37±3.02	33.28±0.13	57.08±0.48	37.63±3.71	38.34±1.11
	Freq	40%	91.67±0.62	19.28±0.66	15.15±0.86	**0.35±0.00**	33.60±1.56	31.08±1.25	59.64±0.87	37.80±0.66	36.07±0.81
	HOPE	50%	**94.19±0.36**	37.86±1.99	18.18±1.48	**10.21±0.41**	29.77±2.08	37.99±1.01	**67.20±0.10**	**48.01±4.04**	42.93±1.44
	REAP	50%	93.94±0.00	**44.19±1.56**	**20.00±1.48**	2.57±1.89	**50.15±1.35**	**46.06±7.36**	58.59±4.28	28.49±7.44	**43.00±3.17**
	MAN	50%	93.69±0.36	41.76±1.25	17.58±2.27	3.85±0.26	45.11±0.88	40.44±0.92	55.62±0.83	34.84±1.60	41.61±1.05
	EAN	50%	91.16±1.29	35.86±0.88	15.76±0.86	0.47±0.03	27.01±0.96	36.25±1.00	59.82±0.11	43.33±0.77	38.71±0.74
	Freq	50%	80.05±0.71	13.55±0.73	16.97±1.71	4.70±0.60	24.12±0.40	31.95±1.16	52.90±0.65	33.66±2.74	32.24±1.09
SB-verif	HOPE	10%	94.44±0.71	42.07±0.58	**20.61±1.71**	**0.35±0.00**	64.77±0.50	**43.81±0.38**	63.37±0.36	62.26±0.99	**48.96±0.66**
	REAP	10%	**95.45±0.62**	39.36±0.75	18.79±0.86	**0.35±0.00**	63.01±0.54	43.00±0.29	**64.33±0.19**	62.80±0.53	48.39±0.47
	MAN	10%	94.44±0.94	**42.32±1.49**	19.39±0.86	**0.35±0.00**	61.87±0.47	43.20±1.16	64.21±0.32	62.04±1.14	48.48±0.80
	EAN	10%	91.92±0.71	39.88±0.13	20.00±0.00	**0.35±0.00**	65.28±0.62	36.66±0.57	62.28±0.32	**64.95±0.42**	47.66±0.35
	Freq	10%	91.16±0.36	34.23±1.44	18.79±0.86	**0.35±0.00**	**66.79±0.45**	37.37±1.79	63.10±0.23	61.13±0.92	46.62±0.76
	HOPE	20%	93.43±0.36	41.53±0.84	**19.39±1.71**	**0.35±0.00**	58.61±0.31	**43.40±1.11**	64.10±0.38	**63.92±1.29**	48.09±0.75
	REAP	20%	**94.44±0.36**	**41.66±1.05**	18.79±0.86	**0.35±0.00**	63.61±0.20	40.39±0.69	63.74±0.18	61.99±0.38	48.12±0.46
	MAN	20%	93.94±0.62	40.42±1.23	18.18±0.00	**0.35±0.00**	**67.66±0.08**	41.56±1.03	**64.40±0.46**	61.45±1.84	**48.50±0.66**
	EAN	20%	84.85±1.24	33.78±0.24	16.97±0.86	**0.35±0.00**	58.39±0.37	34.56±0.83	62.66±0.13	55.43±0.30	43.37±0.50
	Freq	20%	26.01±1.29	13.33±0.81	**19.39±2.27**	**0.35±0.00**	62.18±0.13	34.92±0.52	61.80±0.36	45.75±0.20	32.97±0.70
	HOPE	25%	**93.69±0.94**	**40.20±2.08**	19.39±0.86	**0.35±0.00**	57.97±0.25	38.85±0.32	**66.33±0.41**	**61.67±0.27**	47.31±0.64
	REAP	25%	92.68±0.94	38.00±1.84	19.39±0.86	**0.35±0.00**	62.49±0.47	**39.52±1.14**	64.61±0.47	59.62±0.79	47.08±0.82
	MAN	25%	**93.69±0.36**	39.97±1.58	18.79±0.86	**0.35±0.00**	**67.24±0.38**	39.01±0.44	63.79±0.27	58.82±1.33	**47.71±0.65**
	EAN	25%	83.84±0.36	28.44±1.23	19.39±2.27	**0.35±0.00**	63.49±0.59	32.31±0.47	61.60±0.48	54.14±1.24	42.94±0.83
	Freq	25%	15.40±2.17	8.61±0.64	**20.61±0.86**	**0.35±0.00**	61.20±0.33	33.38±0.07	62.25±0.44	41.56±1.19	30.42±0.71
	HOPE	30%	92.68±0.36	40.05±1.17	**20.61±0.86**	**0.35±0.00**	54.59±0.25	**39.52±1.70**	62.95±0.22	60.38±1.22	46.39±0.72
	REAP	30%	93.43±0.94	**40.19±1.01**	**20.61±0.86**	**0.35±0.00**	60.37±0.82	39.31±0.07	**66.95±0.18**	**63.49±0.27**	**48.09±0.52**
	MAN	30%	**94.70±0.62**	39.30±1.24	20.00±1.48	**0.35±0.00**	**61.30±0.55**	38.70±0.88	65.75±0.08	61.83±0.88	47.74±0.72
	EAN	30%	85.61±0.62	27.13±1.17	18.18±0.00	**0.35±0.00**	58.96±0.53	30.47±0.29	59.46±0.82	45.05±0.20	40.65±0.45
	Freq	30%	10.86±0.36	6.67±0.65	18.18±0.00	**0.35±0.00**	60.38±0.25	34.87±0.96	60.49±0.50	32.20±0.68	28.00±0.42
	HOPE	40%	93.18±0.62	**42.46±0.63**	18.18±1.48	**0.35±0.00**	**57.58±1.01**	34.97±0.82	59.43±0.21	**58.92±0.27**	45.64±0.63
	REAP	40%	91.92±0.94	39.93±0.59	19.39±0.86	**0.35±0.00**	56.97±2.93	**36.45±1.82**	63.47±0.28	57.04±0.90	**45.69±1.04**
	MAN	40%	**93.43±0.71**	40.21±0.37	20.00±0.00	**0.35±0.00**	50.63±0.61	33.28±0.76	**64.05±0.55**	56.02±1.59	44.75±0.57
	EAN	40%	84.34±0.71	23.72±0.23	19.39±1.71	**0.35±0.00**	48.38±0.37	32.98±0.66	59.35±0.48	39.03±0.35	38.44±0.56
	Freq	40%	2.78±1.29	1.87±0.07	**20.61±0.86**	**0.35±0.00**	46.31±3.32	29.60±1.52	61.35±0.26	16.77±5.65	22.46±1.62
	HOPE	50%	**91.92±0.36**	**43.99±1.10**	15.15±0.86	6.49±0.72	42.58±1.36	36.15±0.19	61.90±0.52	**52.96±0.73**	**43.89±0.73**
	REAP	50%	89.65±0.36	40.04±0.76	15.76±0.86	5.93±0.03	38.10±0.48	**38.96±0.78**	60.73±0.47	51.29±0.13	42.56±0.48
	MAN	50%	91.41±0.36	37.89±0.61	**17.58±1.71**	**7.24±0.14**	**48.02±1.17**	36.20±1.64	**63.23±0.48**	28.33±3.35	41.24±1.18
	EAN	50%	51.26±1.99	13.33±0.46	16.36±1.48	0.65±0.13	29.42±0.14	33.33±0.47	59.87±0.31	9.78±0.20	26.75±0.65
	Freq	50%	2.02±0.71	1.32±0.21	16.97±3.43	0.72±0.39	30.89±2.47	38.09±1.45	61.38±0.31	8.39±0.47	19.97±1.18

Table S1: Full Tulu results across all calibration sets and pruning rates for Qwen3.5-35B-A3B. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

Calib	Method	Rate	GSM8K	MATH	IFEval	PopQA	MMLU	BBH	TQA	LCB	Mean
	Base		94.19±0.36	35.67±1.29	19.39±0.86	0.35±0.00	26.68±0.04	36.50±0.45	59.36±0.52	42.58±0.70	39.34±0.53
Evol-CA	HOPE	10%	94.70±0.00	**40.08±0.29**	18.18±0.00	**0.35±0.00**	26.57±0.27	**38.55±1.59**	60.62±0.15	**42.10±0.95**	**40.14±0.41**
	REAP	10%	**94.95±0.36**	38.43±0.51	**19.39±1.71**	**0.35±0.00**	27.59±0.43	37.22±0.51	58.41±0.69	41.08±1.33	39.68±0.69
	MAN	10%	94.70±0.00	36.03±1.63	18.18±0.00	**0.35±0.00**	**27.69±0.16**	36.61±0.91	59.99±0.18	39.41±0.73	39.12±0.45
	EAN	10%	94.44±0.36	36.53±0.54	18.18±1.48	**0.35±0.00**	26.97±0.49	36.76±0.71	62.24±0.24	40.11±3.80	39.45±0.95
	Freq	10%	94.19±0.36	36.54±0.69	18.79±0.86	**0.35±0.00**	26.07±0.64	35.89±0.75	**62.52±0.50**	37.96±0.66	39.04±0.56
	HOPE	20%	**95.71±0.71**	38.49±1.58	18.79±0.86	**0.35±0.00**	25.20±0.26	**38.19±0.70**	57.17±0.41	36.29±0.95	38.77±0.68
	REAP	20%	95.45±0.62	**39.39±2.86**	**19.39±0.86**	**0.35±0.00**	24.28±0.74	36.76±0.36	57.68±0.46	37.96±3.38	38.91±1.16
	MAN	20%	94.44±0.36	38.01±0.84	18.79±0.86	**0.35±0.00**	24.29±0.21	37.83±1.38	60.03±0.37	**42.74±0.26**	39.56±0.53
	EAN	20%	92.42±1.07	37.95±0.81	18.18±0.00	**0.35±0.00**	35.23±0.99	35.48±0.72	60.37±0.44	39.03±0.73	39.88±0.60
	Freq	20%	91.67±1.07	36.36±0.31	16.36±0.00	**0.35±0.00**	**46.81±0.42**	36.76±0.94	**61.14±0.40**	33.98±2.31	**40.43±0.68**
	HOPE	25%	**96.46±0.36**	37.43±2.15	18.18±0.00	**0.35±0.00**	23.82±0.07	37.58±0.50	59.87±0.60	**43.01±0.20**	39.59±0.48
	REAP	25%	94.95±1.56	38.76±1.28	18.79±0.86	**0.35±0.00**	24.84±0.39	37.47±0.64	59.20±0.78	38.98±2.29	39.17±0.97
	MAN	25%	95.20±0.36	**40.59±1.11**	**20.61±0.86**	**0.35±0.00**	25.70±0.04	**37.99±0.32**	57.45±0.41	42.53±1.92	40.05±0.63
	EAN	25%	92.93±0.71	33.06±0.85	18.18±0.00	**0.35±0.00**	44.21±0.41	36.81±1.07	**61.80±0.25**	36.99±1.28	**40.54±0.57**
	Freq	25%	93.69±0.94	33.19±1.61	16.36±0.00	**0.35±0.00**	**45.66±0.92**	36.04±1.03	59.56±0.77	35.81±0.26	40.08±0.69
	HOPE	30%	**96.72±0.36**	38.40±1.53	18.18±0.00	**0.35±0.00**	24.00±1.16	36.45±0.32	57.55±0.25	**44.84±2.77**	39.56±0.80
	REAP	30%	95.45±0.62	**40.62±0.46**	18.18±0.00	**0.35±0.00**	24.84±0.46	34.82±1.25	59.16±2.86	40.86±1.69	39.29±0.92
	MAN	30%	95.45±0.62	38.15±0.70	18.18±0.00	**0.35±0.00**	25.31±0.23	35.99±0.56	58.56±0.16	40.16±1.15	39.02±0.43
	EAN	30%	93.43±0.94	34.04±0.44	**18.79±0.86**	**0.35±0.00**	**50.20±0.46**	34.20±1.30	59.31±0.25	37.20±0.53	**40.94±0.60**
	Freq	30%	93.69±0.71	34.56±1.79	16.36±1.48	**0.35±0.00**	39.56±1.35	**36.81±2.07**	**61.38±0.67**	34.25±4.23	39.62±1.54
	HOPE	40%	94.44±0.36	**43.08±1.20**	**21.82±1.48**	**0.35±0.00**	25.87±0.38	36.86±1.45	56.40±0.33	42.37±1.47	40.15±0.83
	REAP	40%	**96.21±0.62**	40.40±2.81	20.00±0.00	**0.35±0.00**	23.39±0.16	35.84±0.89	56.83±0.68	38.12±0.30	38.89±0.68
	MAN	40%	94.95±0.71	40.66±2.03	20.00±1.48	**0.35±0.00**	23.25±0.30	**39.31±0.14**	58.65±0.30	**55.86±0.55**	41.63±0.69
	EAN	40%	94.70±0.62	39.11±1.68	16.97±0.86	**0.35±0.00**	**50.75±0.76**	37.07±0.47	62.58±0.58	37.20±0.92	**42.34±0.74**
	Freq	40%	93.43±0.71	12.93±0.89	18.79±0.86	**0.35±0.00**	39.36±0.85	36.25±0.59	**63.07±0.32**	45.59±2.11	38.72±0.79
	HOPE	50%	95.20±0.36	42.35±2.78	**21.82±1.48**	**8.90±0.32**	**48.93±3.83**	**50.31±0.99**	55.42±0.78	27.04±2.39	**43.75±1.62**
	REAP	50%	94.70±0.62	**47.29±5.03**	**21.82±0.00**	1.68±1.54	20.87±0.02	38.65±2.73	62.37±2.51	34.09±10.45	40.18±2.86
	MAN	50%	**95.71±0.71**	45.53±0.64	20.00±1.48	7.69±1.15	29.75±0.60	35.99±0.69	**63.70±0.39**	15.00±0.13	39.17±0.72
	EAN	50%	93.94±0.62	36.01±1.51	18.18±2.57	0.35±0.00	40.08±0.47	37.01±1.71	63.52±1.05	**48.92±1.35**	42.25±1.16
	Freq	50%	90.66±0.71	11.13±0.07	18.79±0.86	0.35±0.00	33.40±0.89	38.45±0.75	60.57±0.19	32.26±2.66	35.70±0.77
SB-verif	HOPE	10%	95.71±0.36	38.08±1.54	18.18±0.00	**0.35±0.00**	26.64±0.78	37.58±0.70	62.40±0.15	**44.19±0.95**	**40.39±0.56**
	REAP	10%	94.70±0.00	**39.09±0.32**	18.79±0.86	**0.35±0.00**	**27.95±0.07**	35.38±1.40	60.00±0.40	37.37±1.18	39.20±0.53
	MAN	10%	94.70±0.00	36.66±2.40	**19.39±0.86**	**0.35±0.00**	27.05±0.18	**38.60±0.64**	57.61±0.59	37.26±0.73	38.95±0.68
	EAN	10%	**96.21±0.62**	34.07±2.06	18.79±0.86	**0.35±0.00**	25.89±0.20	33.74±0.22	**62.89±0.72**	39.14±1.65	38.88±0.79
	Freq	10%	94.70±0.62	33.84±0.32	18.79±0.86	**0.35±0.00**	25.17±0.12	33.64±1.00	61.28±0.45	31.83±0.80	37.45±0.52
	HOPE	20%	94.70±1.07	37.03±0.83	**18.79±0.86**	**0.35±0.00**	23.19±0.07	**38.39±1.02**	57.92±0.93	**34.25±1.87**	**38.08±0.83**
	REAP	20%	95.20±0.71	38.25±0.96	**18.79±0.86**	**0.35±0.00**	23.56±0.08	37.42±0.76	54.98±0.19	27.15±0.88	36.96±0.56
	MAN	20%	**95.45±0.00**	**38.95±1.59**	18.18±0.00	**0.35±0.00**	23.51±0.09	35.38±1.53	58.53±0.26	27.90±2.31	37.28±0.72
	EAN	20%	93.18±1.64	32.87±1.53	16.97±0.86	**0.35±0.00**	**25.29±0.38**	35.22±1.22	**65.39±0.24**	28.55±1.83	37.23±0.96
	Freq	20%	90.66±0.94	26.85±1.11	18.18±0.00	**0.35±0.00**	23.08±0.32	35.33±0.71	62.98±0.49	31.51±6.47	36.12±1.26
	HOPE	25%	94.19±0.71	38.86±0.93	**19.39±0.86**	**0.35±0.00**	24.01±0.15	37.58±1.75	57.70±0.46	36.29±1.21	38.55±0.76
	REAP	25%	**95.20±0.36**	37.93±1.50	16.97±0.86	**0.35±0.00**	24.19±0.23	34.56±0.73	54.97±0.56	26.67±1.74	36.36±0.75
	MAN	25%	94.70±0.62	**40.34±1.07**	18.79±0.86	**0.35±0.00**	**25.53±0.08**	36.96±0.76	58.03±0.07	**40.43±0.40**	**39.39±0.48**
	EAN	25%	91.67±0.62	30.62±0.92	18.18±1.48	**0.35±0.00**	23.52±0.28	35.22±1.46	62.97±0.44	27.90±0.26	36.30±0.68
	Freq	25%	34.09±3.44	14.32±0.42	17.58±0.86	**0.35±0.00**	22.32±0.22	**39.47±1.45**	**63.31±0.44**	38.01±0.55	28.68±0.92
	HOPE	30%	94.19±0.36	**41.27±2.07**	18.18±1.48	**0.35±0.00**	26.17±0.07	36.04±0.66	59.95±0.55	33.44±2.45	**38.70±0.96**
	REAP	30%	94.95±0.36	41.26±1.72	**18.79±0.86**	**0.35±0.00**	**26.86±0.09**	35.28±0.66	58.91±0.62	30.16±0.70	38.32±0.63
	MAN	30%	**95.71±0.36**	40.72±1.93	16.97±0.86	**0.35±0.00**	24.84±0.33	37.17±0.92	56.88±0.56	31.13±3.10	37.97±1.01
	EAN	30%	87.88±1.24	26.87±0.08	**18.79±0.86**	**0.35±0.00**	22.64±0.25	33.84±0.19	**66.20±1.39**	**42.80±1.19**	37.42±0.65
	Freq	30%	11.62±4.12	5.60±0.40	**18.79±0.86**	**0.35±0.00**	21.28±0.18	**39.57±0.78**	62.95±0.58	39.57±0.38	24.97±0.91
	HOPE	40%	95.45±0.62	42.59±1.81	18.79±0.86	**0.35±0.00**	25.53±0.35	**38.55±0.51**	59.84±0.72	23.44±1.07	38.07±0.74
	REAP	40%	**96.72±0.36**	**48.51±0.61**	18.79±2.27	**0.35±0.00**	24.41±0.30	34.97±0.22	61.86±0.39	33.33±1.22	39.87±0.67
	MAN	40%	96.21±0.00	47.63±0.98	**20.61±0.86**	**0.35±0.00**	24.25±0.15	34.56±0.19	61.86±0.74	**43.71±0.99**	**41.15±0.49**
	EAN	40%	75.00±1.64	17.85±0.70	18.18±1.48	**0.35±0.00**	21.82±0.40	34.41±1.34	**63.75±0.42**	35.48±0.92	33.36±0.86
	Freq	40%	3.79±1.07	3.05±0.65	16.97±0.86	**0.35±0.00**	**29.22±3.17**	34.41±1.00	59.98±0.85	17.47±6.51	20.66±1.76
	HOPE	50%	93.94±0.62	47.16±1.55	**20.00±0.00**	**0.35±0.00**	23.25±0.78	**45.35±0.91**	**63.97±0.71**	**31.08±2.24**	**40.64±0.85**
	REAP	50%	95.20±0.36	**59.39±1.94**	19.39±2.27	**0.35±0.00**	26.63±0.50	36.55±1.34	63.52±0.14	23.60±0.93	40.58±0.93
	MAN	50%	**95.45±0.62**	48.26±2.15	18.79±0.86	**0.35±0.00**	26.11±0.46	36.30±0.81	63.67±0.28	29.25±0.42	39.77±0.70
	EAN	50%	50.00±3.76	9.26±1.31	19.39±0.86	**0.35±0.00**	**35.91±0.58**	38.29±1.08	60.84±0.51	15.65±1.30	28.71±1.17
	Freq	50%	2.53±0.36	1.27±0.16	16.36±0.00	**0.35±0.00**	33.36±0.20	37.99±0.62	60.91±1.07	13.33±1.51	20.76±0.49

Table S2: Full Tulu results across all calibration sets and pruning rates for Qwen3.5-122B-A10B. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

Calib	Method	Rate	GSM8K	MATH	IFEval	PopQA	MMLU	BBH	TQA	LCB	Mean
	Base		90.91±1.24	33.65±1.49	29.09±1.48	30.20±0.52	32.51±0.21	40.70±1.18	58.86±0.24	58.71±0.26	46.83±0.83
Evol-CA	HOPE	10%	88.89±0.71	32.89±0.93	27.88±3.09	20.56±0.22	30.47±0.03	**41.97±0.81**	58.51±0.58	56.56±0.33	44.72±0.84
	REAP	10%	89.90±0.94	32.25±1.53	27.88±1.71	16.16±0.23	**32.21±0.29**	39.57±0.66	58.94±0.09	57.31±0.68	44.28±0.77
	MAN	10%	90.66±1.29	32.70±0.78	29.09±0.00	18.50±0.47	32.04±0.25	41.10±0.66	56.22±0.94	**59.73±1.41**	**45.01±0.72**
	EAN	10%	**90.91±1.24**	**33.73±0.88**	27.88±0.86	19.83±0.51	31.79±0.21	40.13±0.56	57.70±0.42	56.61±1.05	44.82±0.72
	Freq	10%	85.35±0.94	28.34±0.67	**29.70±0.86**	**21.37±0.40**	30.39±0.36	40.24±1.34	**58.95±0.43**	51.02±0.97	43.17±0.75
	HOPE	20%	88.89±0.71	33.96±0.49	24.85±2.27	8.67±0.27	30.83±0.38	40.08±0.62	53.76±0.55	56.18±2.27	42.15±0.94
	REAP	20%	89.90±0.36	**34.38±0.64**	23.64±1.48	9.20±0.38	32.43±0.18	39.42±0.66	54.38±0.18	54.73±0.27	42.26±0.52
	MAN	20%	90.15±0.00	32.37±0.45	26.67±0.86	9.02±0.12	32.42±0.23	39.47±1.61	52.86±0.27	**57.53±1.32**	42.56±0.61
	EAN	20%	**91.16±0.94**	33.45±1.78	27.27±0.00	13.24±0.69	31.40±0.05	36.96±0.98	56.30±0.53	52.74±1.37	**42.82±0.79**
	Freq	20%	66.92±1.29	15.01±0.22	**27.88±2.27**	**15.39±0.97**	**40.58±0.86**	**43.15±2.51**	**61.02±0.44**	37.69±0.59	38.46±1.14
	HOPE	25%	90.66±1.56	36.60±0.97	24.24±1.71	8.53±0.17	32.22±0.55	41.31±0.81	55.33±0.61	55.59±0.95	**43.06±0.92**
	REAP	25%	89.65±1.29	36.05±0.92	23.64±1.48	8.57±0.37	31.50±0.41	40.75±1.07	51.01±0.79	**55.65±1.19**	42.10±0.94
	MAN	25%	90.40±0.36	33.19±1.51	24.24±0.86	8.60±0.46	31.05±0.20	39.93±0.92	52.42±0.46	54.46±0.38	41.79±0.64
	EAN	25%	**91.16±0.94**	**37.04±2.28**	25.45±0.00	10.60±0.59	31.87±0.07	37.42±0.70	55.14±0.53	45.43±1.71	41.77±0.85
	Freq	25%	61.87±3.78	13.21±0.79	**29.70±0.86**	**13.95±0.15**	**40.45±0.87**	**48.31±1.64**	**59.86±0.57**	30.05±0.62	37.18±1.16
	HOPE	30%	89.90±0.71	37.89±1.78	26.67±1.71	8.46±0.23	31.26±0.15	38.80±1.23	51.44±0.29	49.78±1.10	41.77±0.90
	REAP	30%	**92.68±0.71**	**39.02±0.90**	26.67±3.09	9.18±0.15	30.77±0.06	39.01±0.47	49.97±1.02	48.06±0.47	41.92±0.86
	MAN	30%	89.65±1.29	38.11±0.97	27.27±0.00	9.18±0.11	31.01±0.07	38.75±0.26	53.20±0.68	**51.18±0.20**	**42.29±0.45**
	EAN	30%	91.92±0.36	35.92±1.50	**29.09±1.48**	9.30±0.54	30.77±0.47	38.09±0.69	55.57±0.99	44.41±1.51	41.88±0.94
	Freq	30%	43.94±2.70	9.47±0.48	28.48±2.27	**11.82±0.14**	**35.47±1.51**	**47.70±0.76**	**57.63±1.04**	11.72±0.99	30.78±1.23
	HOPE	40%	90.40±1.29	41.18±1.20	24.24±3.74	7.62±0.62	30.96±0.61	39.37±1.45	53.02±0.33	37.04±2.02	40.48±1.41
	REAP	40%	90.91±1.24	**48.17±0.53**	**27.88±3.09**	8.36±0.49	34.64±0.16	39.47±1.13	55.48±0.91	31.61±2.98	42.07±1.31
	MAN	40%	**91.41±0.36**	45.68±1.30	24.85±2.27	9.02±0.54	34.49±0.39	**39.52±0.71**	57.58±0.59	29.19±0.95	41.47±0.89
	EAN	40%	85.35±1.89	43.81±0.60	23.03±3.09	9.09±0.22	**46.00±0.12**	38.80±0.95	**57.89±0.74**	**40.65±0.80**	**43.08±1.05**
	Freq	40%	16.16±2.34	2.53±0.39	24.24±2.27	**9.62±0.34**	32.23±0.36	37.99±1.01	55.92±1.00	4.73±0.40	22.93±1.01
	HOPE	50%	90.15±1.07	54.30±0.87	21.82±0.00	7.31±0.49	35.72±0.67	43.56±0.70	58.10±1.01	**35.11±0.08**	43.26±0.61
	REAP	50%	**90.66±1.29**	**66.81±0.48**	20.61±0.86	8.57±0.18	**41.39±0.51**	46.32±1.19	59.39±0.40	23.76±0.30	**44.69±0.65**
	MAN	50%	86.87±1.29	58.93±1.07	18.79±1.71	8.27±0.32	37.49±0.00	**48.36±0.14**	**60.03±0.43**	15.54±0.53	41.79±0.69
	EAN	50%	79.29±2.17	35.91±0.74	13.94±0.86	8.85±0.26	38.80±0.28	44.43±0.95	57.82±0.70	31.77±0.92	38.85±0.86
	Freq	50%	5.05±1.79	1.04±0.32	**23.64±1.48**	**13.22±0.33**	29.66±0.32	36.91±1.61	55.03±2.14	0.27±0.20	20.60±1.02
SB-verif	HOPE	10%	89.65±0.36	32.17±1.03	28.48±0.86	16.14±0.18	30.60±0.09	40.75±1.02	57.65±1.03	55.54±1.67	43.87±0.78
	REAP	10%	90.66±0.94	**32.79±1.82**	27.27±0.00	15.02±0.28	31.03±0.06	**41.16±0.38**	55.66±0.36	54.14±1.30	43.47±0.64
	MAN	10%	**90.91±1.07**	32.49±1.34	27.27±1.48	18.85±0.49	**31.63±0.41**	39.31±0.51	55.86±0.18	**55.91±1.53**	**44.03±0.88**
	EAN	10%	87.63±0.36	31.68±0.49	27.27±0.00	19.83±0.11	30.78±0.35	39.88±0.63	56.52±0.22	48.06±1.27	42.71±0.43
	Freq	10%	82.07±0.94	17.62±0.44	**29.09±1.48**	**25.79±0.46**	30.51±0.36	37.93±0.81	**58.08±0.45**	23.39±1.34	38.06±0.79
	HOPE	20%	87.88±1.07	32.77±0.96	22.42±1.71	10.72±0.25	30.28±0.20	39.47±0.51	**54.80±0.49**	**56.61±2.12**	41.87±0.91
	REAP	20%	87.37±1.99	32.62±1.50	26.06±0.86	9.41±0.17	30.27±0.42	38.70±0.81	51.27±0.63	54.89±0.30	41.33±0.84
	MAN	20%	**89.65±0.94**	**33.91±0.80**	25.45±1.48	11.31±0.17	30.32±0.47	**39.57±2.30**	54.35±0.75	54.09±1.24	**42.33±1.02**
	EAN	20%	85.61±1.24	27.46±1.54	26.06±0.86	10.58±0.62	30.27±0.36	38.34±1.09	52.62±0.45	38.06±5.07	38.63±1.40
	Freq	20%	58.33±5.97	6.89±0.48	**28.48±2.27**	**15.74±0.36**	**33.12±1.84**	38.85±1.08	54.60±1.01	10.86±0.33	30.86±1.67
	HOPE	25%	88.64±2.23	**37.61±1.05**	28.48±0.86	7.31±0.26	**32.51±0.61**	38.85±0.62	49.22±1.11	53.12±1.00	41.97±0.97
	REAP	25%	**89.90±0.94**	34.03±0.72	26.06±2.27	8.99±0.37	30.15±0.80	38.85±0.07	53.40±0.64	**55.27±0.40**	42.08±0.78
	MAN	25%	88.38±1.79	33.89±1.29	**29.70±1.71**	8.41±0.51	30.00±0.12	**39.31±0.40**	**57.71±0.85**	55.22±1.33	**42.83±1.00**
	EAN	25%	84.60±0.94	25.15±0.36	28.48±3.09	8.46±0.65	29.82±0.28	38.45±0.94	55.84±1.32	33.01±1.24	37.98±1.10
	Freq	25%	41.67±8.92	4.36±0.91	27.27±1.48	**14.30±0.20**	30.31±1.64	36.91±0.59	54.84±0.99	6.29±1.72	26.99±2.06
	HOPE	30%	88.89±1.99	**38.03±1.32**	28.48±3.09	8.20±0.95	**33.96±0.18**	**40.44±1.09**	54.62±0.47	50.00±0.86	**42.83±1.25**
	REAP	30%	**90.91±1.86**	36.55±0.52	28.48±2.27	7.90±0.20	32.89±0.51	37.73±0.38	54.52±0.50	**50.65±0.92**	42.45±0.89
	MAN	30%	87.12±1.64	34.61±0.39	24.85±0.86	7.15±0.26	29.52±0.35	39.72±0.95	54.61±0.57	48.76±0.99	40.79±0.75
	EAN	30%	80.56±1.29	20.52±0.65	**29.70±2.27**	9.16±0.03	28.99±0.83	39.88±1.63	53.66±1.24	32.20±1.54	36.83±1.19
	Freq	30%	32.07±1.29	2.66±0.32	28.48±1.71	**12.43±0.77**	31.98±0.56	36.04±0.82	**56.95±1.57**	6.18±0.27	25.85±0.91
	HOPE	40%	84.60±0.36	42.93±1.42	**30.30±0.86**	7.45±0.29	**37.15±0.31**	44.84±0.94	57.07±0.84	46.24±1.80	**43.82±0.85**
	REAP	40%	85.10±1.99	**44.66±1.07**	27.27±1.48	7.90±0.29	34.20±0.64	**44.89±0.59**	56.82±0.98	41.02±2.21	42.73±1.16
	MAN	40%	**86.62±1.29**	37.19±0.60	23.03±2.27	7.64±0.30	29.20±0.40	39.62±0.14	**58.74±0.73**	**46.34±0.61**	41.05±0.79
	EAN	40%	74.24±3.76	13.51±1.27	**30.30±2.27**	9.11±0.15	30.86±0.62	42.94±0.45	49.59±1.54	24.73±0.73	34.41±1.35
	Freq	40%	7.58±0.62	1.70±0.12	21.82±1.48	**10.84±0.32**	31.02±1.64	36.81±1.07	55.15±0.60	1.88±0.76	20.85±0.83
	HOPE	50%	77.53±1.79	47.80±2.13	19.39±0.86	**8.85±0.07**	36.58±0.85	49.18±0.29	57.19±0.67	30.65±1.55	40.90±1.02
	REAP	50%	79.29±0.94	**56.36±3.07**	18.79±3.74	8.71±0.43	**43.56±0.43**	**50.31±1.81**	54.97±0.98	35.32±1.65	**43.41±1.63**
	MAN	50%	**80.56±3.52**	47.71±1.74	20.00±1.48	7.62±0.34	31.71±0.41	45.65±1.49	**59.47±0.51**	**36.18±3.22**	41.11±1.59
	EAN	50%	51.52±5.50	10.77±1.24	**28.48±3.09**	8.08±0.86	30.18±0.77	43.05±1.57	56.78±1.28	15.38±0.80	30.53±1.89
	Freq	50%	2.02±0.94	0.83±0.22	18.79±3.09	8.60±0.26	28.92±1.87	36.91±1.26	54.59±0.78	0.05±0.08	18.84±1.06

Table S3: Full Tulu results across all calibration sets and pruning rates for GLM-4.5-Air. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

Method	Evol-CA			SB-verif		
	10%	25%	50%	10%	25%	50%
Base	41.82	41.82	41.82	41.82	41.82	41.82
HOPE	43.18±1.40	**44.58±1.42**	**42.33±1.23**	**43.01±0.22**	42.02±1.03	**40.78±0.86**
REAP	41.45±0.59	42.74±0.55	36.23±0.93	42.43±1.23	41.66±0.64	40.21±0.43
MAN	**43.31±0.94**	42.68±1.69	37.54±1.87	42.44±1.58	**42.52±0.95**	40.62±1.19
EAN	41.51±0.89	42.88±1.56	40.56±0.67	41.76±0.46	38.99±0.87	36.24±0.45
Freq	42.79±0.82	42.54±1.31	34.06±0.26	38.80±1.46	37.71±0.18	30.16±0.94

Table S4: Full SWE-Bench Pro results across all calibration sets and pruning rates for Qwen3.5-35B-A3B. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

Method	Evol-CA			SB-verif		
	10%	25%	50%	10%	25%	50%
Base	50.16	50.16	50.16	50.16	50.16	50.16
HOPE	**52.44±0.68**	**52.29±0.54**	43.84±1.62	**53.22±0.66**	51.39±0.47	50.63±0.21
REAP	49.28±0.46	47.17±0.81	39.61±0.27	51.94±1.26	51.55±0.18	46.78±0.68
MAN	52.17±0.34	47.80±0.55	41.81±0.18	51.11±0.55	**51.81±0.30**	44.92±1.32
EAN	52.10±0.44	51.55±0.27	**48.32±1.62**	50.66±0.83	49.62±0.42	**51.80±0.27**
Freq	51.47±0.59	52.00±0.77	45.71±1.10	50.29±1.25	50.87±1.68	45.62±0.92

Table S5: Full SWE-Bench Pro results across all calibration sets and pruning rates for Qwen3.5-122B-A10B. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

Method	Evol-CA			SB-verif		
	10%	25%	50%	10%	25%	50%
Base	28.70	28.70	28.70	28.70	28.70	28.70
HOPE	32.94±1.05	32.80±0.66	**28.89±0.76**	34.64±0.54	33.26±0.96	**28.94±1.21**
REAP	33.23±0.82	**34.06±0.37**	26.88±1.20	34.46±0.59	33.44±0.40	28.70±1.40
MAN	33.93±1.72	32.91±1.32	23.65±0.90	**34.70±1.07**	**33.46±0.90**	26.73±0.91
EAN	**34.09±0.22**	32.62±0.48	20.88±1.51	33.41±1.11	32.71±0.51	26.11±0.84
Freq	27.40±0.54	1.14±0.62	0.00±0.00	30.53±0.60	22.47±0.92	1.69±0.13

Table S6: Full SWE-Bench Pro results across all calibration sets and pruning rates for GLM-4.5-Air. Scores are multiplied by 100 (mean and standard deviation across 3 independent calibration trials). The best method (per condition) for each column is bolded.

		Calibration			
Model	Method	Time/token (ms)	1k prompts (hr)	QP solve (s)	Ratio
Qwen-35B	HOPE	2.68	1.27	75.4	1.077×
$(L = 40, E = 256)$	REAP	2.53	1.20	N/A	
Qwen-122B	HOPE	3.47	1.55	92.4	1.067×
$(L = 48, E = 256)$	REAP	3.31	1.48	N/A	

Table S7: Computational cost of HOPE vs REAP. Calibration time is given in milliseconds per token, and in hours per 1000 prompts (given a random sample of Evol-CodeAlpaca). QP solve time is the time taken to solve the HOPE-specific quadratic program over all layers total (a one-time cost). Ratio is the total amount of time taken for HOPE to make its pruning decisions (including QP solve time) over the time taken for REAP.

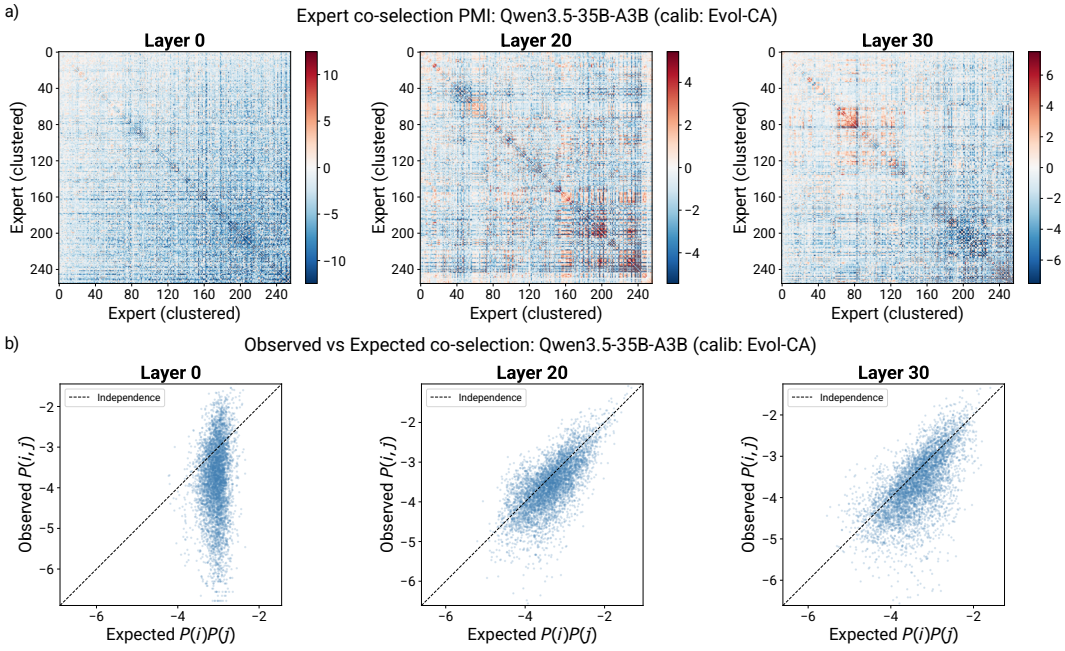


Figure S2: Expert co-selection is highly non-random, which motivates interaction-aware pruning. **a)** Pointwise mutual information (PMI) between expert pairs. PMI is defined between two experts i, j (within a layer) as $PMI(i,j) = log_2 \frac{P(i,j)}{P(i)P(j)}$, where $P(i,j)$ is the fraction of tokens which activate both experts, and $P(i)$ is the fraction of tokens which activate only one expert i. We show PMI between experts at three different layers of Qwen3.5-35B-A3B. Positive values (red) indicate experts are co-selected more often than expected due to independent chance; negative values (blue) indicate experts are co-selected less often (mutual exclusion). **b)** Observed co-selection probability $P(i,j)$ versus independence expectation $P(i)P(j)$. Under completely independent routing, all points should lie on the diagonal. Any substantial deviation (which is especially pronounced at layer 0) demonstrates that experts operate in cooperative groups.

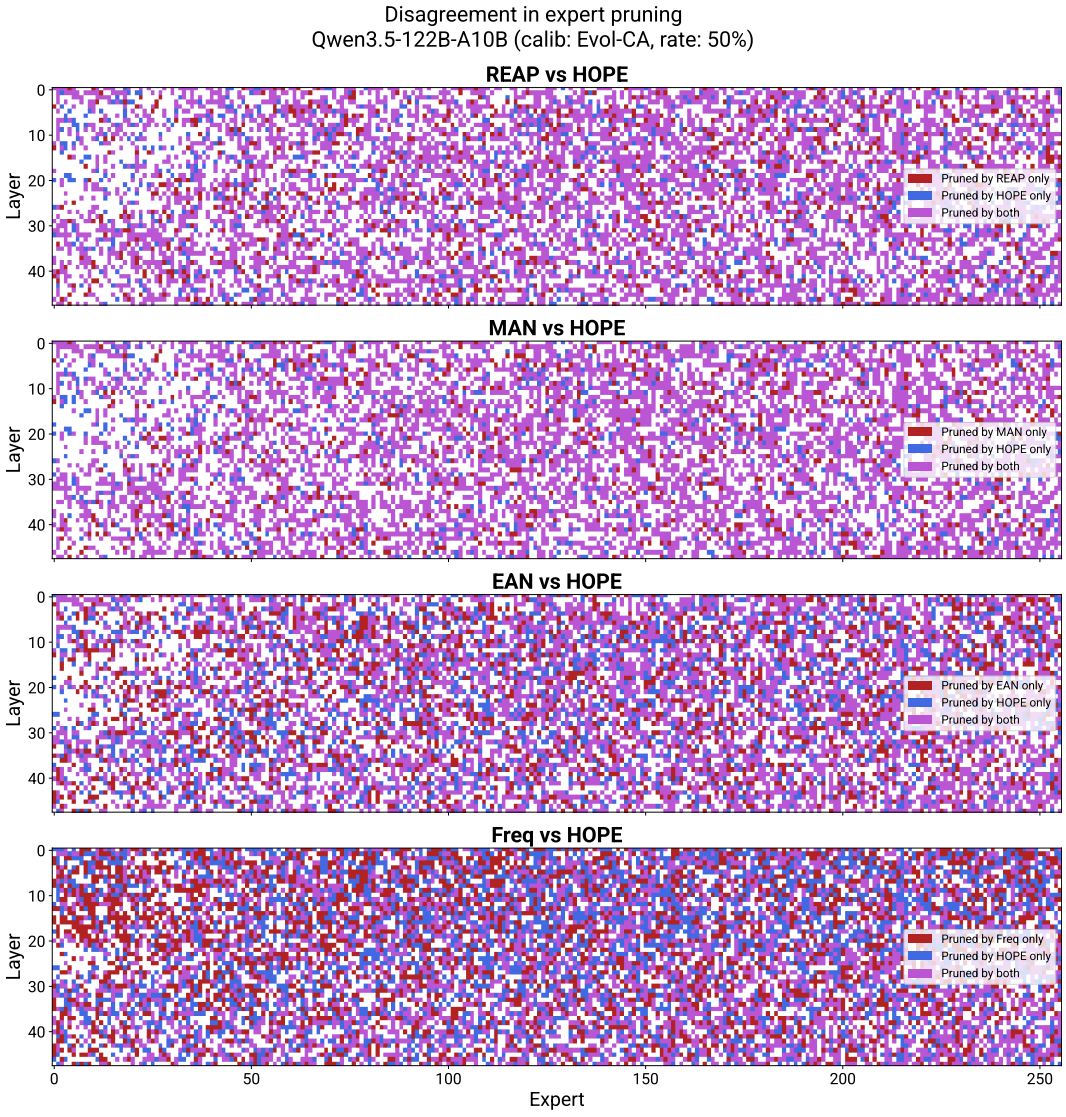


Figure S3: Spatial visualization of disagreement in expert pruning between HOPE and each baseline (Qwen3.5-122B-A10B). Each panel shows an $L \times E$ grid. Disagreement is typically distributed across all layers and experts rather than purely concentrated in certain regions.

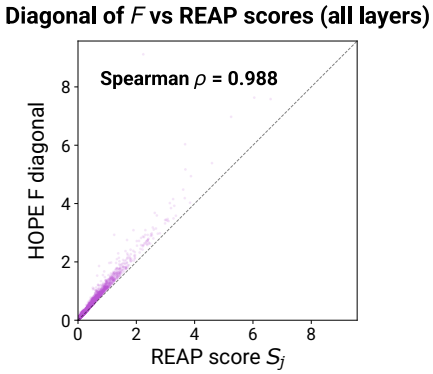


Figure S4: The diagonal of the interaction matrix F recovers REAP's expert rankings. Each point represents one expert (pooled across layers of Qwen3.5-122B-A10B, calib: Evol-CA). The x-axis is the REAP score S_k^{REAP} and the y-axis is $\sqrt{F_{k,k}}$. The Spearman correlation is near perfect ($\rho = 0.988$), confirming that REAP is a special case of HOPE.

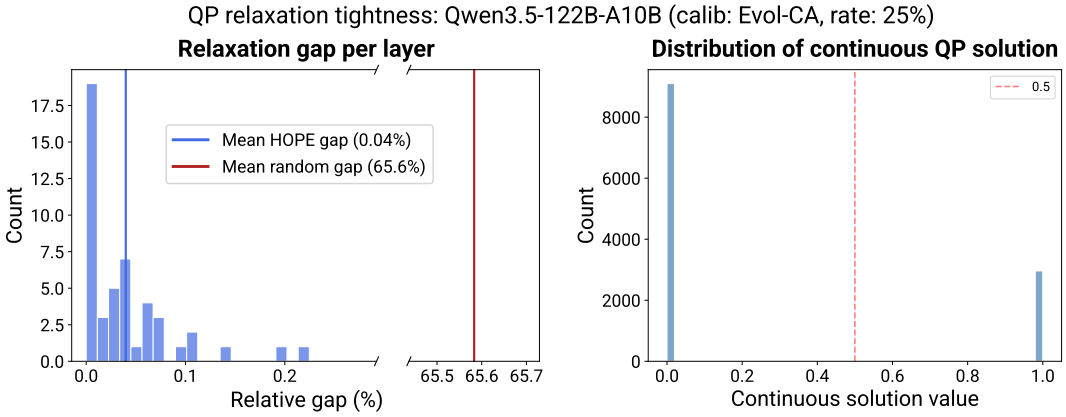


Figure S5: HOPE's continuous relaxation to solve the quadratic program is tight. **a)** Histogram of the relative gap in the binary objective value versus the continuous objective value (measured as $\frac{\text{obj}_{bin} - \text{obj}_{cont}}{\text{obj}_{bin}}$) per layer. The mean gap is only 0.04%, compared to an expected gap of 65.6% for randomly sampled feasible binary solutions (red line). **b)** Distribution of continuous solution values in p across all layers and experts. The overwhelming concentration of continuous solutions near 0 and 1 confirms that the relaxation produces effectively binary solutions, thereby rounding is nearly lossless.

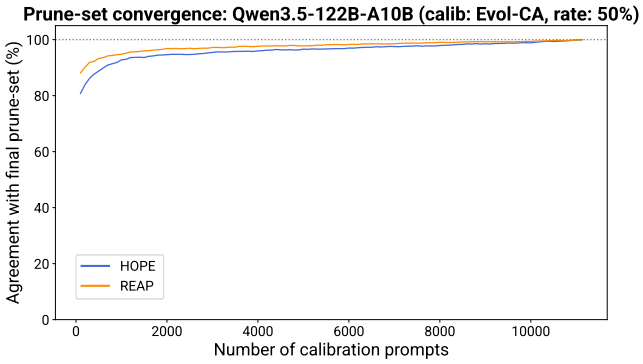


Figure S6: Convergence of pruning decisions with calibration data. We measure the fraction of pruning decisions that agree with the final pruning set (computed from the entire calibration set) as a function of the number of calibration prompts processed. Both HOPE and REAP stabilize within 10k prompts, with HOPE requiring marginally more data to converge due to estimating pairwise (rather than per-expert) statistics. Convergence is shown for 50% pruning on Qwen3.5-35B-A3B with the Evol-CodeAlpaca calibration set.

C CROSS-LAYER HOPE (CHOPE)

In this section, we present an extension of HOPE which accounts for expert interactions across different layers, which enables non-uniform layer-wise pruning budgets (i.e. optimizing for a different number of pruned experts in each layer). We call this extension CHOPE (**C**ross-layer **H**igher-**O**rder **P**runing of **E**xperts).

C.1 SETUP

Consider an L-layer MoE transformer, where each layer ℓ consists of attention followed by an MoE layer, each with residual connections:

$$z^\ell = x^{\ell-1} + \text{Att}^\ell(x^{\ell-1})$$
$$x^\ell = z^\ell + \text{MoE}^\ell(z^\ell)$$

where $x^{\ell-1} \in \mathbb{R}^d$ is the residual stream entering layer ℓ, and $z^\ell \in \mathbb{R}^d$ is the post-attention representation fed to the MoE. Layer norms are absorbed into the respective Att, MoE functions. Note that in this derivation, we assume each layer consists of attention followed by an MoE block, but the attention can be equivalently replaced by other architectures (e.g. an SSM).

A forward pass computes $x^0, z^1, x^1, \ldots, x^L$ in sequence, where x^0 is the input token embedding and x^L is fed to the language-model head.

At each layer ℓ, the MoE contains E experts with output $\text{MoE}^\ell(z^\ell) = \sum_{k=1}^{E} g_k^\ell(z^\ell)\, f_k^\ell(z^\ell)$, using the same notation as in Appendix A. We use top-K routing with selected set $T^\ell(z^\ell) \subseteq \{1, \ldots, E\}$.

At each layer ℓ, we prune a set of experts $P^\ell \subseteq \{1, \ldots, E\}$. Upon pruning, for any token, the pruned experts $P^\ell \cap T^\ell(z^\ell)$ are replaced by the next-highest-logit experts $R^\ell(z^\ell)$, with $|R^\ell(z^\ell)| = |P^\ell \cap T^\ell(z^\ell)|$. The pruned MoE output uses new gate weights $g'^\ell_k(z^\ell)$. Note that this is the same setup as in Appendix A, tracking layer indices.

C.2 ERROR PROPAGATION

When we prune experts, the effects of pruning propagate through the network from early to later layers. After pruning experts from each layer, the forward pass computes perturbed quantities $\tilde{x}^0, \tilde{z}^1, \tilde{x}^1, \ldots, \tilde{x}^L$ (with $\tilde{x}^0 = x^0$). Define the output *error* at layer ℓ as $\epsilon^\ell = \tilde{x}^\ell - x^\ell$. Here, we will quantify the error ϵ^ℓ in each layer.

Error through attention: At layer ℓ, the perturbed post-attention representation is $\tilde{z}^\ell = \tilde{x}^{\ell-1} + \text{Att}^\ell(\tilde{x}^{\ell-1})$. The error in the output of this attention layer is:

$$\tilde{z}^\ell - z^\ell = (\tilde{x}^{\ell-1} - x^{\ell-1}) + (\text{Att}^\ell(\tilde{x}^{\ell-1}) - \text{Att}^\ell(x^{\ell-1}))$$

Since no pruning/modification happens at the attention block, its error $\tilde{z}^\ell - z^\ell$ arises purely from error in the input being propagated through.

To quantify $\tilde{z}^\ell - z^\ell$, we perform a first-order Taylor expansion of Att^ℓ around $x^{\ell-1}$:

$$\text{Att}^\ell(\tilde{x}^{\ell-1}) \approx \text{Att}^\ell(x^{\ell-1}) + J_A^\ell(\tilde{x}^{\ell-1} - x^{\ell-1})$$

where $J_A^\ell = \frac{\partial\, \text{Att}^\ell(x)}{\partial x}\big|_{x=x^{\ell-1}} \in \mathbb{R}^{d\times d}$ is the Jacobian of the attention output with respect to its input, evaluated at the unperturbed input.

Substituting, we have:

$$\tilde{z}^\ell - z^\ell \approx (I + J_A^\ell)\, \epsilon^{\ell-1}$$

Error through MoE: Similar to error propagation through attention, the error in the output of the MoE is:

$$\tilde{x}^\ell - x^\ell = (\tilde{z}^\ell - z^\ell) + (\widetilde{\text{MoE}}^\ell(\tilde{z}^\ell) - \text{MoE}^\ell(z^\ell))$$

where $\widetilde{\text{MoE}}^{\ell}$ is the MoE block after pruning. Unlike the attention block, the error in the output of the MoE block arises from both propagated error *and* the error accrued from pruning experts within the block.

Let $\Delta^{\ell}(\tilde{z}^{\ell}) = \widetilde{\text{MoE}}^{\ell}(\tilde{z}^{\ell}) - \text{MoE}^{\ell}(\tilde{z}^{\ell})$ be the error resulting from pruning at layer ℓ (the difference in MoE outputs before and after pruning, applied to the perturbed input). Substituting, this gives us:

$$\tilde{x}^{\ell} - x^{\ell} = (\tilde{z}^{\ell} - z^{\ell}) + (\text{MoE}^{\ell}(\tilde{z}^{\ell}) - \text{MoE}^{\ell}(z^{\ell})) + \Delta^{\ell}(\tilde{z}^{\ell})$$

We perform a first-order Taylor approximation of the unpruned MoE around z^{ℓ} (like with attention), yielding:

$$\epsilon^{\ell} = \tilde{x}^{\ell} - x^{\ell} \approx (I + J_M^{\ell})\,(\tilde{z}^{\ell} - z^{\ell}) + \Delta^{\ell}(\tilde{z}^{\ell})$$

where $J_M^{\ell} = \frac{\partial\, \text{MoE}^{\ell}(z)}{\partial z}\big|_{z=z^{\ell}}$ is the MoE Jacobian (evaluated at the unperturbed input).

Closed form of the error: Substituting in the attention error $\tilde{z}^{\ell} - z^{\ell}$ into the MoE error ϵ^{ℓ}:

$$\epsilon^{\ell} = (I + J_M^{\ell})(I + J_A^{\ell})\,\epsilon^{\ell-1} + \Delta^{\ell}(\tilde{z}^{\ell})$$

with $\epsilon^0 = 0$.

This recurrence has the following closed form:

$$\epsilon^{L} = \sum_{j=1}^{L} \Phi_{L,j}\,\Delta^{j}(\tilde{z}^{j})$$

where $\Phi_{\ell,j}$ is the *error propagation operator*, defined as follows:

$$\Phi_{\ell,j} = \prod_{m=j+1}^{\ell} (I + J_M^{m})(I + J_A^{m}), \quad \text{with} \quad \Phi_{\ell,\ell} = I$$

$\Phi_{\ell,j}$ quantifies how error introduced at layer j is amplified or attenuated as it propagates to the output at layer ℓ.

Note that ϵ^{L} is the overall error of the last layer of the network, which includes error arising from expert pruning in all MoE layers, as well as the error propagating from earlier to later layers.

Decoupling error from $\tilde{z}^{j}$: Computing $\Delta^{j}(\tilde{z}^{j})$ exactly requires knowing the perturbed input $\tilde{z}^{j}$, which itself depends on all previous pruning decisions. This makes optimizing this error over prune-sets intractable. Instead, we approximate $\Delta^{j}(\tilde{z}^{j}) \approx \Delta^{j}(z^{j})$. That is, the pruning error at layer j is well-approximated by the error computed on the clean (unpruned) input. This decouples the per-layer pruning errors from the propagation operators:

$$\epsilon^{L} \approx \sum_{j=1}^{L} \Phi_{L,j}\,\Delta^{j}(z^{j})$$

C.3 Upper-bounding the substitution error

Here, we will bound the substitution-error component of ϵ^{L}. The proof follows analogously to that in Appendix A.

As with per-layer HOPE (Appendix A), we decompose $\Delta^{j}(z^{j})$ into a substitution error and a renormalization error:

$$\Delta^{j}(z^{j}) = \left[\sum_{i \in R^{j}(z^{j})} g_i^{'j}(z^{j}) f_i^{j}(z^{j}) - \sum_{k \in T^{j}(z^{j}) \cap P^{j}} g_k^{j}(z^{j}) f_k^{j}(z^{j})\right] + \left[\sum_{k \in T^{j}(z^{j}) \setminus P^{j}} (g_k^{'j}(z^{j}) - g_k^{j}(z^{j})) f_k^{j}(z^{j})\right]$$

Substituting this decomposition into ϵ^L:

$$\epsilon^L = \underbrace{\sum_{j=1}^{L} \Phi^{L,j} \left[\sum_{i \in R^j(z^j)} g_i^{'j}(z^j) f_i^j(z^j) - \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j) f_k^j(z^j) \right]}_{\text{substitution error}} + \underbrace{\sum_{j=1}^{L} \Phi^{L,j} \left[\sum_{k \in T^j(z^j) \setminus P^j} (g_k^{'j}(z^j) - g_k^j(z^j)) f_k^j(z^j) \right]}_{\text{renormalization error}}$$

The renormalization error remains small by the same argument as in Appendix A (with an additional factor of $\Phi_{L,j}$). As before, we focus on minimizing the squared cross-layer substitution error:

$$\left\| \sum_{j=1}^{L} \Phi_{L,j} \Big(\sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, f_k^j(z^j) - \sum_{i \in R^j(z^j)} g'^j_i(z^j)\, f_i^j(z^j) \Big) \right\|_2^2$$

As before, we decompose this squared norm into three components ($A + B + C$):

$$A = \left\| \sum_{j=1}^{L} \Phi_{L,j} \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, f_k^j(z^j) \right\|_2^2$$

$$B = -2 \left\langle \sum_{j=1}^{L} \Phi_{L,j} \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, f_k^j(z^j),\ \sum_{j=1}^{L} \Phi_{L,j} \sum_{i \in R^j(z^j)} g'^j_i(z^j)\, f_i^j(z^j) \right\rangle$$

$$C = \left\| \sum_{j=1}^{L} \Phi_{L,j} \sum_{i \in R^j(z^j)} g'^j_i(z^j)\, f_i^j(z^j) \right\|_2^2$$

By Cauchy–Schwarz, $|B| \le 2\sqrt{A}\sqrt{C}$, so $A + B + C \le (\sqrt{A} + \sqrt{C})^2$.

We now upper bound both $\sqrt{A}$ and $\sqrt{C}$ using a common quantity. Define the following:

$$Z = \Big(\sum_{j=1}^{L} \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \Big)^2$$

Bounding $\sqrt{A}$: By the triangle inequality (applied first over layers, then over experts within each layer):

$$\sqrt{A} = \left\| \sum_{j=1}^{L} \Phi_{L,j} \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, f_k^j(z^j) \right\|_2 \le \sum_{j=1}^{L} \sum_{k \in T^j(z^j) \cap P^j} g_k^j(z^j)\, \|\Phi_{L,j}\, f_k^j(z^j)\|_2 = \sqrt{Z}$$

Bounding $\sqrt{C}$: By the triangle inequality:

$$\sqrt{C} = \left\| \sum_{j=1}^{L} \Phi_{L,j} \sum_{i \in R^j(z^j)} g'^j_i(z^j)\, f_i^j(z^j) \right\|_2 \le \sum_{j=1}^{L} \sum_{i \in R^j(z^j)} g'^j_i(z^j)\, \|\Phi_{L,j}\, f_i^j(z^j)\|_2$$

We have $\|\Phi_{L,j}\, f_i^j(z^j)\|_2 \le \max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2$ for each replacement expert i in each layer j. Therefore:

$$\sum_{j=1}^{L} \sum_{i \in R^j(z^j)} g'^j_i(z^j) \|\Phi_{L,j}\, f_i^j(z^j)\|_2 \le \max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \sum_{j=1}^{L} \sum_{i \in R^j(z^j)} g'^j_i(z^j)$$

By Lemma 3 (applied independently at each layer j), $\sum_{i\in R^j(z^j)} {g'}_i^j(z^j) \le \sum_{k\in T^j(z^j)\cap P^j} g_k^j(z^j)$. Summing over layers:

$$\max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \sum_{j=1}^{L} \sum_{i\in R^j(z^j)} {g'}_i^j(z^j) \le \max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \sum_{j=1}^{L} \sum_{k\in T^j(z^j)\cap P^j} g_k^j(z^j)$$

Furthermore, for each pruned expert k in layer j, we have $g_k^j(z^j) \le g_k^j(z^j) \frac{\|\Phi_{L,j}\, f_k^j(z^j)\|_2}{\min_{j,k} \|\Phi_{L,j} f_k^j(z^j)\|_2}$. Substituting:

$$\max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \sum_{j=1}^{L} \sum_{k\in T^j(z^j)\cap P^j} g_k^j(z^j)$$

$$\le \frac{\max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2}{\min_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2} \sum_{j=1}^{L} \sum_{k\in T^j(z^j)\cap P^j} g_k^j(z^j) \|\Phi_{L,j}\, f_k^j(z^j)\|_2 = \rho_\Phi \sqrt{Z}$$

Thus, $\sqrt{C} \le \rho_\Phi \sqrt{Z}$.

Together:

$$\left\| \sum_{j=1}^{L} \Phi_{L,j} \left(\sum_{k\in T^j(z^j)\cap P^j} g_k^j(z^j)\, f_k^j(z^j) - \sum_{i\in R^j(z^j)} {g'}_i^j(z^j)\, f_i^j(z^j) \right) \right\|_2^2 \le (\sqrt{A} + \sqrt{C})^2$$

$$\le (\sqrt{Z} + \rho_\Phi \sqrt{Z})^2$$

$$= (1+\rho_\Phi)^2 \cdot Z$$

This gives us the following Theorem, analogous to Theorem 1:

Theorem 4 (Cross-layer upper bound)**.** *For any collection of prune-sets $\{P^1, \dots, P^L\}$, the squared cross-layer substitution error is bounded by:*

$$(1+\rho_\Phi)^2 \cdot Z \quad \textit{where} \quad \rho_\Phi = \frac{\max_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2}{\min_{j,k} \|\Phi_{L,j}\, f_k^j(z^j)\|_2}$$

Note that ρ_Φ generalizes the per-layer ρ from Theorem 1: it accounts for how $\Phi_{L,j}$ differentially amplifies expert contributions from different layers. Because the ratio is independent of the prune-sets $\{P^1, \dots, P^L\}$, we can simply minimize Z.

C.4 Minimizing Z as a quadratic program

Our goal is to minimize $\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[Z]$ over all prune-sets $\{P^1, \dots, P^L\}$ jointly. We introduce binary decision variables $p_k^j \in \{0,1\}$, where $p_k^j = 1$ iff expert k in layer j is pruned.

Substituting:

$$Z = \left(\sum_{j=1}^{L} \sum_{k=1}^{E} p_k^j \cdot \mathbf{1}[k \in T^j(z^j)] \cdot g_k^j(z^j)\, \|\Phi_{L,j}\, f_k^j(z^j)\|_2 \right)^2$$

$$= \sum_{j=1}^{L} \sum_{\ell=1}^{L} \sum_{k=1}^{E} \sum_{m=1}^{E} p_k^j\, p_m^\ell \cdot \mathbf{1}[k \in T^j(z^j)] \cdot \mathbf{1}[m \in T^\ell(z^\ell)]$$

$$\cdot\, g_k^j(z^j)\, g_m^\ell(z^\ell)\, \|\Phi_{L,j}\, f_k^j(z^j)\|_2\, \|\Phi_{L,\ell}\, f_m^\ell(z^\ell)\|_2$$

Taking the empirical expectation over the calibration set (using conditional normalization as in per-layer HOPE):

$$\mathop{\mathbb{E}}_{x}[Z] = \sum_{j=1}^{L} \sum_{\ell=1}^{L} \sum_{k=1}^{E} \sum_{m=1}^{E} p_k^j\, p_m^\ell \cdot G[(j,k),(\ell,m)] = p^\top G\, p$$

where $p \in \{0,1\}^{LE}$ is the concatenated binary vector over all layers, and $G \in \mathbb{R}^{LE \times LE}$ is defined as:

$$G[(j,k),(\ell,m)] = \frac{1}{|\mathcal{X}^{j,\ell}_{k,m}|} \sum_{x \in \mathcal{X}^{j,\ell}_{k,m}} g^j_k(z^j)\, g^\ell_m(z^\ell)\, \|\Phi_{L,j}\, f^j_k(z^j)\|_2\, \|\Phi_{L,\ell}\, f^\ell_m(z^\ell)\|_2$$

with $\mathcal{X}^{j,\ell}_{k,m} = \{x : k \in T^j(z^j) \text{ and } m \in T^\ell(z^\ell)\}$ denoting the set of tokens which activate expert k in layer j and expert m in layer ℓ.

Theorem 5 (Cross-layer QP)**.** *Given a total pruning budget B (the total number of experts to remove across all layers), the collection of prune-sets $\{P^1, \ldots, P^L\}$ which minimizes $\mathop{\mathbb{E}}_{x \in \mathcal{D}}[Z]$ is the solution to:*

$$p^* = \arg\min_p \; p^\top G\, p \quad \textit{s.t.} \quad p \in \{0,1\}^{LE},\; \sum_{k=1}^{LE} p_k = B$$

C.5 Relationship to per-layer HOPE

The cross-layer matrix G encodes both within-layer and across-layer expert interactions:

- $E \times E$ blocks on the diagonal of G (entries where $j = \ell$) correspond to the per-layer F-matrices of HOPE, scaled by $\|\Phi_{L,j}\|$
- The off-diagonal blocks (entries where $j \neq \ell$) encode *cross-layer* interactions: the joint penalty of pruning expert k in layer j and expert m in layer ℓ simultaneously

CHOPE strictly generalizes per-layer HOPE. Setting $\Phi_{L,j} = I$ for all j and zeroing the off-diagonal blocks of G (i.e. $G[(j,k),(\ell,m)] = 0$ for $j \neq \ell$) decouples layers entirely, recovering L independent per-layer HOPE problems (Theorem 2). Further zeroing the within-layer off-diagonal entries effectively recovers REAP.

Unlike per-layer HOPE (which enforces a fixed budget per layer), CHOPE optimizes a single global budget $B = \sum_\ell |P^\ell|$ across all layers. The optimization allocates pruning non-uniformly, potentially pruning more aggressively in layers with weaker cooperative structure or lower propagation impact.

C.6 Practical considerations

Jacobian approximation: Computing the full $d \times d$ Jacobians J^ℓ_A and J^ℓ_M at every layer is prohibitively expensive for large models. A practical simplification is to set $\Phi_{L,j} = I$ for all j, corresponding to the assumption that errors propagate through the residual stream without transformation (i.e. Jacobians are approximately zero). For residual-stream architectures where the skip connection dominates, this is a reasonable approximation. Under this simplification, G retains cross-layer interaction terms but does not differentially weight layers based on error propagation.

Per-layer minimum constraints: In practice, optimizing a global budget without constraints can lead to instability (e.g. over-pruning early layers). We propose adding per-layer minimum constraints to ensure at least M experts (the routing width) survive in every layer:

$$p^* = \arg\min_p \; p^\top G\, p \quad \text{s.t.} \quad p \in \{0,1\}^{LE},\; \sum_{k=1}^{LE} p_k = B,\; \sum_{k=1}^{E} (1 - p^j_k) \geq M \;\; \forall\, j$$

Scalability: The matrix G has dimension $LE \times LE$. While larger than the per-layer $E \times E$ matrix, this remains tractable for modern QP solvers.

C.7 Preliminary results

Here, we present some preliminary results comparing the performance of CHOPE to layer-wise HOPE.

Method	Rate	Tulu Mean	SWE-Bench Pro
Base		48.87	41.82
HOPE	10%	**48.49**$\pm$**0.53**	**43.18**$\pm$**1.40**
CHOPE	10%	46.17$\pm$0.72	40.87$\pm$1.41
HOPE	25%	**48.51**$\pm$**0.61**	**44.58**$\pm$**1.42**
CHOPE	25%	42.36$\pm$1.08	27.29$\pm$17.51
HOPE	50%	42.93$\pm$1.44	**42.33**$\pm$**1.23**
CHOPE	50%	**48.12**$\pm$**2.44**	22.95$\pm$1.19

Table S8: HOPE vs cross-layer HOPE (CHOPE) for Qwen3.5-35B-A3B (calib: Evol-CA). Scores $\times 100$ (mean and standard deviation across 3 independent calibration trials). The better method for each column is bolded.

Method	Rate	Tulu Mean	SWE-Bench Pro
Base		39.34	50.16
HOPE	10%	**40.14**$\pm$**0.41**	**52.44**$\pm$**0.68**
CHOPE	10%	39.45$\pm$0.76	46.04$\pm$0.32
HOPE	25%	**39.59**$\pm$**0.48**	**52.29**$\pm$**0.54**
CHOPE	25%	38.72$\pm$1.06	47.10$\pm$0.46
HOPE	50%	**43.75**$\pm$**1.62**	**43.84**$\pm$**1.62**
CHOPE	50%	40.30$\pm$0.88	42.40$\pm$0.11

Table S9: HOPE vs cross-layer HOPE (CHOPE) for Qwen3.5-122B-A10B (calib: Evol-CA). Scores $\times 100$ (mean and standard deviation across 3 independent calibration trials). The better method for each column is bolded.

We implemented CHOPE and computed non-uniform prune-sets for Qwen3.5-35B-A3B and Qwen3.5-122B-A10B. For simplicity, we assume that Jacobians are I (as described above), and do not limit the number of experts pruned per layer.

CHOPE's non-uniform allocation is consistent across trials, and in many conditions, CHOPE performed reasonably well, although not as well as per-layer HOPE (Tables S8–S9). CHOPE-pruned models also occasionally experienced performance collapse.

CHOPE tended to prune a large number of experts in the earliest layers of the network (Figure S7), suggesting that the cross-layer objective's pruning allocation can introduce instability. Although promising, limiting the number of experts pruned (particularly in early layers) could help CHOPE perform better and avoid performance collapse. We leave further exploration of CHOPE for future work.

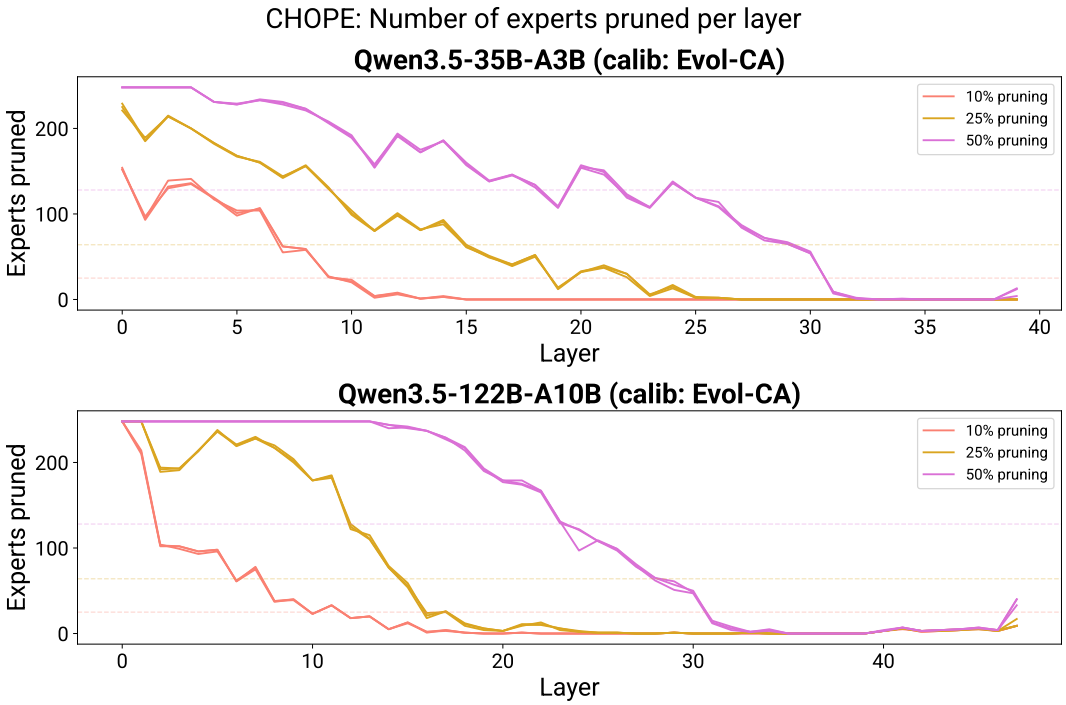


Figure S7: Non-uniform expert pruning allocation by CHOPE (cross-layer HOPE). Each line indicates the number of experts pruned per layer (3 independent trials for each pruning rate); dashed horizontal lines indicate the uniform budget (used by per-layer HOPE).

D METHODS

D.1 CALIBRATION

Expert usage statistics for HOPE were collected via a single forward pass over the calibration set. We hooked into each MoE layer's expert module and—for each token—recorded: 1) which experts were in the top-K selected set, 2) their softmax-normalized gate weights, and 3) their output vectors. From these, we accumulated per-layer statistics in an $E \times E$ matrix: the total gate-weighted norm products $\sum_{x:x\in\mathcal{X}_{i,j}} g_i(x)g_j(x)\|f_i(x)\|\|f_j(x)\|$ for all co-selected expert pairs i, j, along with co-selection counts $|\mathcal{X}_{\rangle,|}|$.

We used two calibration sets:

- **Evol-CodeAlpaca-v1** (Luo et al., 2023): A coding-focused instruction dataset. We used all 111k prompts on the unpruned model.
- **SWE-bench Verified trajectories** (Jimenez et al., 2023): Agentic multi-turn coding trajectories collected from the unpruned model solving SWE-Bench Verified instances (500 instances). Full trajectories (including tool calls and observations) were reconstructed and tokenized.

For each calibration set (and each model), we ran 3 independent trials using the same prompts but different random seeds for model generation, yielding different routing patterns and thus different F matrices per trial. For SWE-bench Verified, we truncated any traces longer than 100k tokens (rare).

D.2 QUADRATIC-PROGRAM SOLVER

Given the $E \times E$ interaction matrix F for a layer, we solved the binary QP $\min_p p^\top Fp$ subject to $p \in \{0, 1\}^E$ and $\sum_k p_k = |P|$ via continuous relaxation. We relaxed $p \in \{0, 1\}^E$ to $p \in [0, 1]^E$ and solved with SciPy's SLSQP optimizer, using bounds $[0, 1]$ per variable and a hard equality constraint on the sum. The initial point was set to $p_k = \frac{|P|}{E}$ (uniform). We used a tolerance of 10^{-12} and a maximum of 1000 iterations. The continuous solution was then rounded to binary by selecting the $|P|$ entries with the highest values.

For first-order baselines (REAP, MAN, EAN, Frequency), we computed their respective scalar scores using the same calibration data and collected the statistics as described in their respective publications, and sorted to select the $|P|$ experts with the lowest scores.

D.3 MODEL PRUNING

Pruned model checkpoints were created by physically removing expert parameters. For each pruned expert, we deleted its MLP weights (`gate_up_proj` and `down_proj`) and sliced out the corresponding rows from the router's weight matrix. This reduced the model's parameter count and file size proportionally. The pruned model was saved as a standard HuggingFace checkpoint and loaded directly for inference.

D.4 TULU3 EVALUATION

Pruned models were evaluated on the Tulu3-Dev benchmark suite using the OLMo Evaluation Suite (olmes) (Lambert et al., 2024). We evaluated on 8 tasks: GSM8K, MATH, IFEval, MMLU, BBH, TruthfulQA, PopQA, and LiveCodeBench. We dropped Codex from the suite due to insufficient variation between models (LiveCodeBench is a more realistic set of coding tasks). Models were served via vLLM with a maximum sequence length of 8192 tokens and a generation budget of 2048 tokens per instance for computational efficiency. These are the same parameters we used for Evol-CodeAlpaca calibration on the unpruned models.

D.5 SWE-BENCH PRO EVALUATION

Agentic coding evaluation was conducted on SWE-Bench Pro (Deng et al., 2025), a benchmark of 731 enterprise-level software engineering instances. We used the Harbor evaluation framework with

the OpenHands SDK agent (`openhands-sdk-bench`). The pruned model was served via vLLM and accessed by the agent as a hosted endpoint. We used the following evaluation parameters: `n_attempts=1`, `max_input_tokens=253952`, `max_output_tokens=8192`, temperature 0.6, top-p 0.95. The metric reported is the mean reward (resolve rate) across all 731 instances. These are the same parameters we used for SWE-Bench verified calibration on the unpruned models.

D.6 SUPERVISED FINE-TUNING

Post-pruning SFT was conducted on Qwen3.5-122B-A10B pruned checkpoints (HOPE and REAP, at 50% pruning rate, Evol-CodeAlpaca calibration). We fine-tuned on LiveCodeBench traces collected from the unpruned model, tokenized with sequence parallelism (8-way). Training used DeepSpeed ZeRO Stage 3 with full parameter fine-tuning (not LoRA), BF16 mixed precision, and Flash Attention 2. We used the following learning hyperparameters: learning rate 10^{-6}, constant schedule with 20-step warmup, max gradient norm 1.0, global batch size $\sim$2M tokens/step (per-device batch size 1, gradient accumulation every 2 steps, sequence parallel size 8), trained for 500 steps ($\sim$1B tokens total). Pre-SFT and post-SFT evaluations used matched calibration trials for fair comparison.