Pruning and Distilling Mixture-of-Experts into Dense Language Models

Junhyuck Kim
KRAFTON

Jihun Yun
KRAFTON

Haechan Kim*
KAIST

Gyeongman Kim
KRAFTON

Joonghyun Bae
KRAFTON

Jaewoong Cho
KRAFTON

Abstract

Mixture-of-Experts (MoE) is now the dominant architecture for frontier language models, yet it requires all expert parameters to be loaded in memory, making it less preferable for memory-constrained deployment. Existing compression methods reduce the number of experts but the output remains an MoE model with the same fundamental limitation. We present the first systematic framework for converting a trained MoE into a standard *fully dense* architecture: experts are scored, selected, and grouped, then concatenated into a dense FFN and refined by knowledge distillation from the MoE teacher. We evaluate 7 scoring, 5 grouping, and 2 magnitude scaling methods across a range of selected expert counts on Qwen3-30B-A3B, yielding 350 configurations. We find that the choice of scoring method is the most impactful, with our novel diversity-aware scoring consistently outperforming prior methods on Qwen3-30B-A3B, DeepSeek-V2-Lite, and GPT-OSS-20B. Under a controlled comparison at matched parameter count, MoE-to-dense outperforms dense-to-dense pruning by $+6.3$ pp in average downstream accuracy after $\sim$4B-token distillation at $1.6\times$ faster training wall-clock speed.

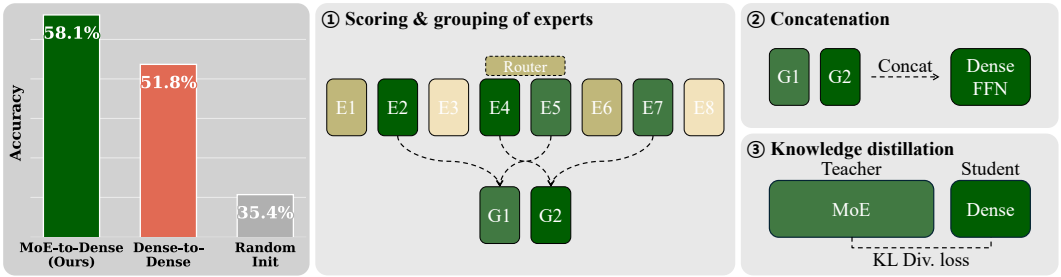


Figure 1: **Left:** Average downstream accuracy (Section 3.1) after $\sim$4B-token distillation, comparing three approaches to producing a 3B-parameter dense student. *MoE-to-dense* uses the MoE teacher (Qwen3-30B-A3B) to initialize and distill the student; *Dense-to-dense* prunes a dense teacher of matched total parameter count (Qwen3-32B); *Random Init* trains the same architecture from scratch via distillation. **Right:** The MoE-to-dense pipeline. From the original MoE layer, we score and select a subset, group and merge them into k groups, then concatenate into a standard dense FFN with appropriate magnitude scaling. We evaluate 7 scoring, 5 grouping, and 2 magnitude scaling methods to identify the best configuration.

*Work done during an internship at KRAFTON.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: MoE compression methods and their design choices. Prior work produces smaller MoE models, while ours is the first to target a fully dense architecture. See Appendix A for a detailed comparison.

Output	Method	Scoring	Grouping	Distill
MoE	MC-SMoE [24]	Selection freq	Router-logit anchor-based	✓
	HC-SMoE [5]	Selection freq	Output-sim agglomerative	–
	Sub-MoE [23]	Selection freq	Output-sim K-means	–
	MergeMoE [29]	Selection freq	Weight-sim anchor-based	–
	PuzzleMoE [48]	Weight × activation	Entry-wise diff	–
	REAP [22]	Gate × act. mag.	–	–
Dense	**This work**	**7 methods** (best: gate × act. mag. + diversity)	**5 methods**	✓

1 Introduction

Mixture-of-Experts (MoE) [12, 38] has become the dominant architecture for the most capable models from LLM providers [11, 28, 45, 47] as it enables scaling the total number of parameters while remaining practical to train and serve. However, MoE models require loading all expert parameters into memory despite activating only a fraction per token, making them less preferable for memory-constrained scenarios such as on-device deployment or single-GPU serving. To address these deployment needs, many providers release families of small dense models alongside their flagship MoE [13, 26, 45]. These dense models are typically either trained from scratch or produced through cascaded pruning and distillation of a large dense teacher, which must itself first be trained [26, 31, 39]. Both approaches require substantial compute and do not leverage the capability present in the MoE model, which, as we show, provides a stronger starting point for distillation than a dense teacher of matched parameter count.

Recent work has demonstrated that MoE models can be compressed into a smaller MoE by reducing the number of experts (Table 1). While highly effective, the resulting models still require loading all remaining experts into memory, preserving the fundamental memory inefficiency. Although individual methods can be adapted to produce dense outputs, no prior work has systematically studied MoE-to-dense conversion (Appendix A discusses related work in detail).

To fill this gap, we introduce the first systematic framework for converting a trained MoE language model into a standard *fully dense* architecture. Our approach proceeds in three steps: (1) experts are scored by importance, and the top-scoring experts are selected, grouped, and merged within groups, (2) the merged experts are concatenated into a dense FFN, with down-projection matrices scaled to preserve output magnitude, and (3) knowledge distillation [16] from the MoE teacher recovers quality lost during compression. Figure 1 illustrates the pipeline.

To identify the best recipe, we evaluate 7 different expert scoring methods, including our novel diversity-aware scoring metric, 5 different grouping methods, 2 different down-projection scaling options, and a wide range of the number of selected experts on Qwen3-30B-A3B, resulting in 350 combinations. Below are our key findings, which are further validated across DeepSeek-V2-Lite and GPT-OSS-20B:

1. **Expert scoring is the dominant factor.** Best and worst scoring methods differ by 5.7 pp in average downstream accuracy, while grouping strategy contributes only $\sim$1 pp (Section 3.2).
2. **Selecting diverse experts without merging is the best strategy.** On all three models, the best configuration uses diversity-aware scoring with pure pruning (no weight averaging). In contrast, frequency-based scoring selects redundant experts and benefits from merging them (Section 3.2, 3.5).
3. **MoE-to-dense outperforms dense-to-dense pruning.** In a controlled comparison at matched total parameter count ($\sim$30B teacher, 3B student), our best configuration outperforms dense-to-dense pruning [31] by $+6.3$ pp after $\sim$4B-token distillation, and is $1.6\times$ faster in training wall-clock time (Section 3.2, 3.3).

Table 2: MoE expert dimensions across models used in this work. The active FFN width per token ($d_{\text{expert}} \times k$) falls in the typical dense range of $\sim$3–5 $\times\, d$.

Model	d	d_{expert}	E	k	$d_{\text{dense}}{=}d_{\text{expert}}{\times}k$	d_{dense}/d
Qwen3-30B-A3B	2048	768	128	8	6144	3.0×
GPT-OSS-20B	2880	2880	32	4	11520	4.0×
DeepSeek-V2-Lite	2048	1408	64	6	8448*	4.1×

*DeepSeek-V2-Lite also has 2 shared (always-on) experts; the full dense equivalent is $2{\times}1408 + 6{\times}1408 = 11264$.

Our contributions are:

1. **First MoE-to-dense pruning and distillation framework.** We present the first end-to-end study of converting a trained MoE language model into a fully dense architecture.
2. **Diversity-aware expert selection via Gram log-determinant.** We introduce a D-optimal selection criterion that maximizes the log-determinant of an importance-weighted Gram matrix, jointly capturing expert importance and mutual diversity. Combined with activation-weighted scoring, it achieves the best accuracy across all configurations and all three models.
3. **Comprehensive evaluation and cross-model validation.** We evaluate 350 configurations and identify the best recipe under 0.3B-token distillation. The advantage persists under extended training ($\sim$4B tokens) and generalizes to DeepSeek-V2-Lite and GPT-OSS-20B.

2 Method

We present a method for converting a Mixture-of-Experts (MoE) language model into a dense model with a total number of parameters equivalent to the active number of parameters of the teacher MoE model. Given E experts per MoE layer with k routed per token, the method selects the top-K experts by importance ($K \geq k$), assigns them to k groups (copying directly when $K = k$, or merging via score-weighted averaging when $K > k$), and concatenates the resulting weights into a standard dense FFN. Knowledge distillation from the MoE teacher then recovers quality lost during compression. Throughout this section, k denotes the MoE router's top-k count (an architecture constant), while K denotes the number of routed experts we select for conversion (a design choice).

2.1 MoE-to-dense conversion

Consider an MoE transformer with L layers, each containing a router that maps a hidden representation $\mathbf{h} \in \mathbb{R}^d$ to logits over E experts. The router selects the top-k experts per token and computes a weighted combination of their outputs. We consider the case where each expert is a gated MLP (SwiGLU [37]) with intermediate dimension d_{expert}, parameterized by $\mathbf{W}^{(e)}_{\text{gate}}, \mathbf{W}^{(e)}_{\text{up}} \in \mathbb{R}^{d_{\text{expert}} \times d}$ and $\mathbf{W}^{(e)}_{\text{down}} \in \mathbb{R}^{d \times d_{\text{expert}}}$. MoE architectures typically size experts so that $d_{\text{expert}} \times k$ falls in the typical dense FFN range of $\sim$3–5 $\times\, d$ (Table 2), keeping the active computation per token comparable to a dense model.

We convert each MoE layer into a dense FFN by first scoring all E experts by importance and retaining the top-K. How experts are scored is detailed in Section 2.2; we evaluate seven methods and find that the choice of scoring is the most impactful decision. When $K > k$, the selected experts are assigned to k groups and merged within each group via score-weighted averaging (grouping strategies are compared in Section 2.3). Let $s^{(e)}$ denote the importance score of expert e and $\mathcal{G}_g$ the set of experts assigned to group g:

$$\mathbf{W}^{(g)}_{\text{proj}} = \sum_{e \in \mathcal{G}_g} \frac{s^{(e)}}{\sum_{e' \in \mathcal{G}_g} s^{(e')}} \mathbf{W}^{(e)}_{\text{proj}}, \quad \text{proj} \in \{\text{gate}, \text{up}, \text{down}\}. \tag{1}$$

When $K = k$, each group contains exactly one expert and no merging is needed.

The k resulting weight matrices are then concatenated into a single dense FFN with intermediate dimension $d_{\text{dense}} = k \times d_{\text{expert}}$:

$$\begin{aligned} \mathbf{W}_{\text{gate}} &= \left[\mathbf{W}^{(1)}_{\text{gate}};\ \ldots;\ \mathbf{W}^{(k)}_{\text{gate}}\right] \in \mathbb{R}^{d_{\text{dense}} \times d}, \quad \mathbf{W}_{\text{up}} = \left[\mathbf{W}^{(1)}_{\text{up}};\ \ldots;\ \mathbf{W}^{(k)}_{\text{up}}\right] \in \mathbb{R}^{d_{\text{dense}} \times d}, \\ \mathbf{W}_{\text{down}} &= \left[\tilde{\mathbf{W}}^{(1)}_{\text{down}},\ \ldots,\ \tilde{\mathbf{W}}^{(k)}_{\text{down}}\right] \in \mathbb{R}^{d \times d_{\text{dense}}}, \end{aligned} \tag{2}$$

where $[\,\cdot\,;\,\cdot\,]$ denotes row-concatenation, $[\,\cdot\,,\,\cdot\,]$ column-concatenation, each $\mathbf{W}_{\text{proj}}^{(g)}$ is the weight matrix for group g, and $\tilde{\mathbf{W}}_{\text{down}}^{(g)} = \alpha_g \mathbf{W}_{\text{down}}^{(g)}$ is the down-projection scaled to approximate the average routing behavior (Section 2.4). Block concatenation preserves the intermediate activations of the constructed group representatives exactly (Appendix B); when $K{=}k$, these representatives are copied experts, while for $K{>}k$ they are parameter-averaged proxies. The final aggregation still differs from the original MoE because static α_g cannot simulate token-dependent router weights. Attention layers, embeddings, and layer norms are copied unchanged from teacher to student.

The parameter K controls a prune/merge tradeoff: $K{=}k$ copies experts directly (pure pruning), $K{>}k$ averages K/k experts per group, and $K{=}E$ retains all experts (pure merging). The full algorithmic pseudocode is provided in Algorithm 1 in Appendix C.

2.2 Scoring

Expert scores determine which experts to retain (top-K selection) and how to weight them during merging. We evaluate seven methods spanning three families, all computed from statistics collected in the calibration forward pass. For each token t and layer ℓ, the router produces a distribution $p_\ell^{(e)}(t)$ via softmax over all E experts, then selects $\mathcal{S}_\ell(t) = \text{top-k}\{p_\ell^{(e)}(t)\}_{e=1}^{E}$.

Frequency-based scoring (SF, PP, PS). Prior work [5, 24, 29] universally uses **selection frequency (SF)**, the fraction of tokens for which expert e is among the top-k, as the importance metric. We additionally evaluate **pre-selection probability (PP)**, the average softmax probability over all tokens regardless of selection, and **post-selection probability (PS)**, the average over all tokens, taking the softmax probability when e was selected and zero otherwise. All three favor generalist experts that appear frequently. Formal definitions are in Appendix D.

Conditional probability (CP). The frequency-based scores above conflate how often an expert is chosen with how confident the router is when it chooses. CP isolates the latter as the average softmax probability over only the tokens where e was selected:

$$s_\ell^{(e)} = \frac{\sum_{t:e\in\mathcal{S}_\ell(t)} p_\ell^{(e)}(t)}{\left|\{t : e \in \mathcal{S}_\ell(t)\}\right|}. \tag{3}$$

CP is not diluted by frequency, so specialist experts that are rarely selected but confidently routed score highly. As shown in Figure 2, CP selects a very different top-8 set of experts from the frequency-based methods.

Activation-weighted conditional probability (ACP). Routing probabilities ignore the magnitude of expert outputs. Following REAP [22], which scores experts by the product of routing weight and output norm, we define a factorized variant: $s_\ell^{(e)} = \bar{p}_\ell^{(e)} \cdot \sqrt{\mathbb{E}_t[\|f_e(t)\|^2]}$, where $\bar{p}_\ell^{(e)}$ is the conditional probability from Eq. 3 and $f_e(t)$ is the output of expert e on token t. This factorization decouples routing confidence from output magnitude, producing a per-expert scalar that serves as the base importance for the D-optimal selection criterion we introduce next.

D-Optimal selection (DO). The methods above rank experts independently, ignoring redundancy among high-scoring experts. We introduce a D-optimal criterion that jointly maximizes importance and diversity. Given base importance $I_e = s^{(e)}$ (instantiated as CP or ACP) and the expert output Gram matrix $\mathbf{G}_{ij} = \mathbb{E}_t[\langle f_i(t), f_j(t)\rangle]$, we form the importance-weighted kernel $\mathcal{K}_{ij} = \sqrt{I_i I_j} \cdot \mathbf{G}_{ij}$ and select:

$$S^* = \arg\max_{|S|=K} \log\det(\boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\,\mathbf{I})\,, \tag{4}$$

where $\lambda_{\text{reg}} = \frac{1}{KE}\sum_{e=1}^{E}\mathcal{K}_{ee}$. Since exhaustive search over $\binom{E}{K}$ subsets is intractable, we solve this greedily: starting from $S = \emptyset$, we iteratively add the expert that maximally increases $\log\det(\boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\mathbf{I})$, computed efficiently via Schur complement evaluations in $O(K^3E)$ total time (Algorithm 2 in Appendix C). Applying DO to the two best base scores yields **DO-CP** and **DO-ACP**; DO-ACP is the overall best method (Section 3.2). We measure the diversity of a selected expert set by its effective rank [36]: the number of significant independent directions in the expert output space,

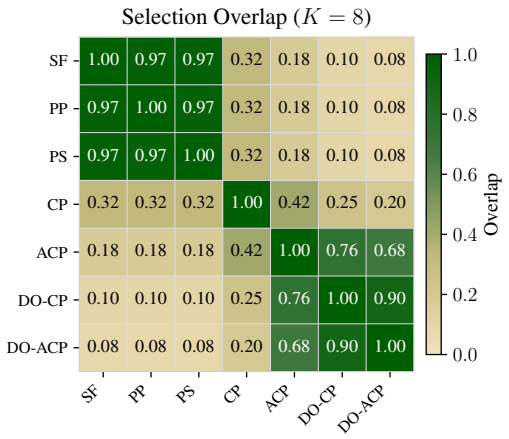


Figure 2: Selection overlap between the top-K=8 experts chosen by each scoring method, averaged across all layers of Qwen3-30B-A3B. Frequency-based methods (SF, PP, PS) select nearly identical experts, while DO-ACP shares $\leq$0.08 overlap with them.

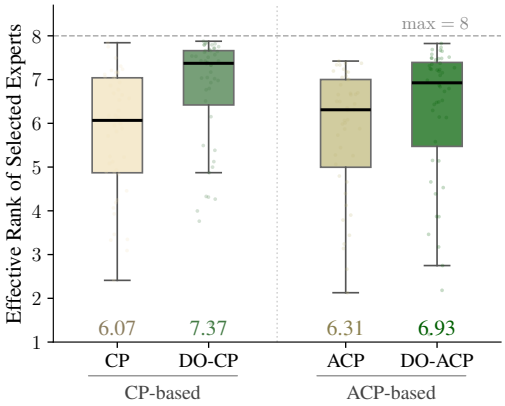


Figure 3: Effective rank [36] of the K=8 selected expert kernel submatrix (max 8) for Qwen3-30B-A3B. D-Optimal selection increases effective rank from 6.07 to 7.37 (CP) and 6.31 to 6.93 (ACP).

ranging from 1 (all experts are near-duplicates) to K (maximally diverse). Figure 3 confirms that D-Optimal selection substantially increases the effective rank compared to independent scoring, avoiding redundant experts as predicted by Theorem 2.1. Excluded scoring methods are discussed in Appendix E.

Theoretical properties. The log-determinant objective in Eq. 4 has three useful properties (full statements and proofs in Appendix F).

Theorem 2.1 (Independent scoring can fail under redundancy)**.** *There exists a family of MoE layers, calibration distributions, and a positive regularization choice for which selecting the top-K experts by ACP incurs a constant reconstruction error, while a size-K subset maximizing the log-determinant objective achieves zero error.*

If several high-scoring experts are near-duplicates, selecting all of them wastes dense capacity. Appendix F.1 constructs such a failure mode explicitly for ACP and shows that the log-determinant objective can avoid it.

Beyond this failure mode, the greedy algorithm we use is provably near-optimal: for a positive semidefinite kernel and fixed $\lambda_{\text{reg}} > 0$, the normalized log-determinant objective $\log\det(\mathbf{I}+\lambda_{\text{reg}}^{-1}\boldsymbol{\mathcal{K}}_S)$ is monotone submodular, so greedy achieves the standard $(1-1/e)$ approximation [32] (Appendix F.2). And when the expert kernel is incoherent (off-diagonal entries small), the diagonal proxy $\sum_{e\in S}\log(\mathcal{K}_{ee}+\lambda_{\text{reg}})$ approximates the full objective up to a coherence-controlled gap (Appendix F.3); this supports the intuition that diversity corrections matter most when redundant experts create large off-diagonal interactions. This is consistent with the cross-model trend in Section 3.5, where Qwen3 (128 experts) shows a large D-Optimal gain while GPT-OSS (32 experts) shows a compressed scoring gap. Additional theoretical results (finite-sample calibration, grouping, merging) are in Appendix G.

2.3 Grouping

After selecting the top-K experts, we assign them to k groups. We evaluate five strategies: **round-robin (RR)**, which assigns score-sorted experts to groups cyclically, producing balanced groups by construction; **weight clustering (WC)** and **router clustering (RC)**, which cluster experts by weight or router-vector similarity; **anchor-based (AB)** [24], which uses the k highest-scoring experts as anchors and assigns the rest by router-vector similarity; and **output clustering (OC)** [5], which clusters on hidden-state outputs. Full definitions are in Appendix H.

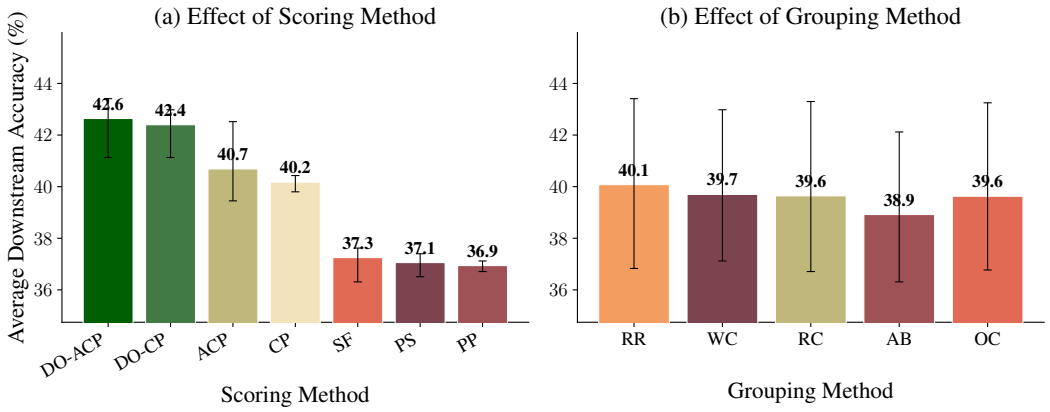


Figure 4: Marginal effect of each design axis on post-distillation downstream accuracy (0.3B tokens). **(a)** Scoring: DO-ACP dominates (42.64% mean), with a 5.7 pp gap to PP (36.94%). **(b)** Grouping: small effect ($\sim$1 pp spread); round-robin (40.08%) narrowly leads.

2.4 Down-projection scaling

In the original MoE, the router dynamically weights each expert's output per token. The concatenated dense FFN has no router, so we apply static scaling factors α_g to each group's down-projection to approximate the average routing behavior. We consider **uniform** scaling ($\alpha_g = 1/k$, splitting routing mass equally across the k groups) and **proportional** scaling ($\alpha_g \propto \sum_{e \in \mathcal{G}_g} s^{(e)}$, weighting each group by its share of selected-expert importance). Full equations are in Appendix I.

3 Experiments

We validate our pipeline on Qwen3-30B-A3B, converting the 30B-parameter MoE into a 3.3B dense model. Section 3.1 describes the setup and experimental workflow. Section 3.2 evaluates every possible scoring$\times$grouping combination after knowledge distillation. Section 3.3 reports extended training, and Section 3.4 covers broader benchmarks, calibration sensitivity, and deployment. Section 3.5 validates findings on DeepSeek-V2-Lite and GPT-OSS-20B. Distillation-method ablations and a qualitative error analysis are in Appendices J and K.

3.1 Experimental setup

Workflow. Our evaluation proceeds in three stages. First, we sweep all combinations of 7 scoring methods, 5 grouping strategies, 2 down-projection (DP) scalings, and $K \in \{8, 16, 32, 64, 128\}$, yielding 350 configurations. Each is evaluated by WikiText-2 perplexity before distillation (we refer to it as pre-distill PPL), measuring initialization quality. Second, for each of the 35 scoring$\times$grouping pairs, we select the best $(K, \text{DP scaling})$ by pre-distill PPL and distill for 0.3B tokens. Third, all distilled models are evaluated on five downstream benchmarks.

Models. Our primary teacher is **Qwen3-30B-A3B** [45], a 30B-parameter MoE with E=128 experts and k=8 active per token. We use the post-trained variant. (We find that the main findings are consistent when using the base model as the teacher model, see Appendix L for a detailed comparison.) Regardless of the teacher variant, the student is a base model, distilled on a generic web corpus without instruction tuning. The dense student has 3.3B parameters with $d_{\text{dense}} = k \times d_{\text{expert}} = 6{,}144$. For cross-model validation (Section 3.5), we additionally test **DeepSeek-V2-Lite** [9, 10] and **GPT-OSS-20B** [1]; full specs are in Table 2.

Training. Expert importance is calibrated on 512 WikiText-103 [27] sequences of up to 2048 tokens. Unless otherwise noted, distillation minimizes forward KL between teacher and student output logits per token, on FineWeb-Edu [34] with global batch size 384, sequence length 4096,

Table 3: Comparison to baselines on Qwen3-30B-A3B (0.3B-token distillation). DO-ACP at $K=8$ uses our diversity-aware scoring; SF×AB and SF×OC use the metrics of MC-SMoE [24] and HC-SMoE [5] repurposed for a dense student. D2D pruning is Dense-to-dense pruning of a matched-parameter dense teacher (Qwen3-32B). Random FFN + teacher attn copies the teacher's attention with a random FFN. Random initialization trains the same architecture from scratch.

Configuration	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
DO-ACP, $K=8$ (best)	**57.0**	**41.1**	**57.4**	**29.9**	**31.7**	**43.41**
SF×AB (MC-SMoE metrics)	52.2	31.6	47.1	23.1	27.5	36.31
SF×OC (HC-SMoE metrics)	53.1	33.1	49.3	24.7	27.3	37.52
D2D pruning (Qwen3-32B → 3.4B)	51.7	27.7	38.9	22.4	25.7	33.28
Random FFN + teacher attn	53.4	28.0	34.7	22.2	25.2	32.70
Random initialization	51.2	25.3	28.9	22.4	23.0	30.15
Teacher (Qwen3-30B-A3B)	82.6	82.0	85.3	65.3	83.2	79.65
D2D Teacher (Qwen3-32B)	76.8	84.0	89.9	73.1	81.9	81.14

learning rate 10^{-4} with cosine decay to 10^{-5}, and AdamW (β_1=0.9, β_2=0.95). Full distillation setup, loss/routing ablations, hyperparameters, and infrastructure details are in Appendices J and M.

Evaluation. Following Muralidharan et al. [31], Sreenivas et al. [39], downstream accuracy is measured on Winogrande (5-shot), HellaSwag (10-shot), ARC-Easy (25-shot), ARC-Challenge (25-shot), and MMLU (5-shot). We report the unweighted average across all five tasks.

3.2 Scoring and grouping evaluation

From the 350-configuration pre-distill PPL sweep (see Appendix N for the complete 350-configuration pre-distill PPL grid), the best K for each scoring×grouping pair is overwhelmingly $K=8$ or $K=16$ (32 of 35 configs, only 3 select $K\geq 32$), indicating that retaining a small number of high-quality experts is more effective than merging many. We distill all 35 best scoring×grouping combinations for 0.3B tokens each and evaluate on five downstream benchmarks (Figure 4). The full 35-combination accuracy matrix with per-benchmark breakdown is in Appendix O (Table 15).

Specialist selection and diversity are crucial. Three distinct tiers emerge across the 35-combination grid (Figure 4), with a 5.7 pp gap between the best and worst scoring families, roughly $5\times$ the grouping spread (1.2 pp). At the bottom tier ($\sim$37%), frequency-based methods (SF, PP, PS) favor generalist experts that serve many tokens but capture broadly shared features; they select nearly identical expert sets (Figure 2), explaining their tight clustering. The middle tier (40–41%) shows that *focusing on rare-but-confident experts matters*: CP removes the frequency factor, promoting specialists that are rarely selected but confidently routed, yielding $+3$ pp. ACP further adds output magnitude information ($+0.5$ pp). The top tier (42–43%) demonstrates that *diversity among selected experts matters*: DO-ACP and DO-CP apply the D-optimal criterion to explicitly penalize redundancy, gaining another $+2$ pp. This progression, from generalist to specialist to diverse specialist, demonstrates that routing confidence and inter-expert diversity provide complementary signals for expert selection, with additive gains when combined.

Grouping matters less when few experts are selected. The best-performing configurations all use small K (8 or 16), which limits the role of grouping (Appendix N). At $K=8$ ($= k$), each group contains exactly one expert, making grouping completely irrelevant; all five strategies produce identical results. At $K=16$, pre-distill PPL varies by up to $\sim 3\times$ across grouping methods within a scoring family (ACP 2,002–6,334, CP 2,148–6,033; Table 15), but distillation largely compensates and the post-distill PPL spread collapses to within 3 perplexity points. Across the 35 best-per-row configurations, the grouping-induced accuracy spread remains small: RR leads at 40.08%, the three clustering methods trail by only $\sim$0.4 pp, and AB underperforms by $\sim$1 pp (Figure 4).

Pure pruning outperforms merging for diversity-aware scoring. The top-7 configurations all use $K=8$ (pure pruning: one expert per group, no weight averaging; see Table 15). Although $K>8$ yields lower pre-distill PPL, distillation reverses the ranking: ACP $K=8$ reaches 42.52% post-distill

Table 4: Extended training results after ∼4B tokens. The ranking established at 0.3B tokens holds at scale: DO-ACP maintains its advantage, reaching 58.10% average accuracy, +6.3 pp over D2D and +12.7 pp over random FFN.

Configuration	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
DO-ACP, K=8	**63.1**	**60.3**	**75.6**	**45.4**	**46.1**	**58.10**
SF, K=16	61.2	56.3	74.0	43.1	32.7	53.46
D2D pruning (Qwen3-32B → 3.4B)	60.5	57.5	73.1	41.5	26.6	51.84
Random FFN + teacher attn	54.4	45.4	66.0	34.2	27.1	45.44
Qwen3-1.7B (pretrained reference)	66.1	67.1	81.9	55.5	62.6	66.63
Qwen3-4B (pretrained reference)	72.0	75.8	86.2	64.6	73.1	74.34

vs. 40.50% for ACP×OC K=16, a +2.0 pp gap. The same holds with the base teacher (Appendix L); cross-model results reveal a more nuanced interaction between scoring and K (Section 3.5). DO-ACP at K=8 achieves the highest average accuracy across all 35 configurations (43.41%; Table 15). Since K=8 dominates and grouping is irrelevant at K=k, we refer to the best configuration simply as DO-ACP hereafter.

Comparison to baselines. We find that MC-SMoE's [24] (SF×AB, 36.31%) and HC-SMoE's [5] (SF×OC, 37.52%) scoring×grouping metrics, evaluated within our framework, lag DO-ACP by 7.1 pp and 5.9 pp (Table 3). Beyond our framework, the strongest comparison is dense-to-dense (D2D) pruning: following the Minitron approach [31], we search over five student architectures at matched parameter count (∼3.4B), select the best by pre-distill PPL, and distill it with its dense teacher (Qwen3-32B) using identical hyperparameters and token budget (Appendix P). With this setup, D2D reaches only 33.28%, 10.1 pp below DO-ACP. D2D is also barely above the *random FFN* baseline (32.70%, +0.6 pp), which copies the teacher's attention layers but initializes FFN weights randomly; this gap suggests that one-shot dense pruning provides little structural advantage at this compression ratio. A full random initialization baseline (30.15%) confirms that expert structure provides a much stronger initialization than random weights. We also strengthen the D2D baseline with iterative prune-and-distill schedules (Appendix P): the strongest schedule reaches 40.20%, which DO-ACP leads by +3.2 pp and on every benchmark.

3.3 Extended training

All results above use 0.3B-token distillation. To investigate whether our findings persist at scale, we train four configurations for ∼4B tokens. The four configurations are: **DO-ACP** K=8 (our best), **SF** K=16 with weight clustering (best selection-frequency configuration from Section 3.2), **D2D** (dense-to-dense pruning baseline), and **Random FFN** (random FFN + teacher attention).

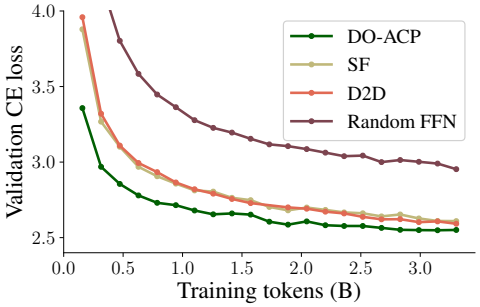


Figure 5: Validation CE loss during extended training. DO-ACP maintains the lowest loss throughout.

The ranking established at 0.3B tokens holds at scale. DO-ACP reaches **58.10%** average accuracy, outperforming D2D pruning (51.84%) by +6.3 pp, SF (53.46%) by +4.6 pp, and random FFN (45.44%) by +12.7 pp (Table 4). The gap is especially pronounced on MMLU: DO-ACP reaches 46.1% versus 32.7% (SF), 26.6% (D2D), and 27.1% (random), suggesting that diversity-aware expert selection may particularly benefit knowledge-intensive tasks during extended training. Figure 5 confirms that DO-ACP maintains the lowest validation CE loss throughout training. We provide a qualitative analysis of these checkpoints, including a taxonomy of failure modes, knowledge-error rates, and a subject-level breakdown, in Appendix K.

MoE-to-dense distillation is also faster than dense-to-dense (D2D) pruning: 73 s/step vs. 116 s/step on identical hardware (2×B200 GPUs, GBS=384), a **1.6× speedup**. This is due to the more efficient

Table 5: Broader benchmarks after ∼4B-token distillation. BPB columns are bits-per-byte (↓ better); GSM8K, RULER, and Belebele are accuracy in % (↑ better). **Bold**: best per column.

	Code BPB		Math		Instr.	Long-ctx	Multilingual		Agentic
Model	**HumanEval**	**MBPP**	**GSM8K**	**BPB**	**BPB**	**RULER**	**Belebele**	**BPB**	**BPB**
DO-ACP	0.703	**1.139**	**12.05**	**0.856**	**0.791**	**89.94**	**35.01**	1.323	**0.780**
SF	0.837	1.441	2.43	0.980	0.809	80.86	26.19	1.334	0.859
D2D	**0.695**	1.404	2.43	0.945	0.809	80.09	25.99	**1.322**	0.817
Random FFN	1.039	1.621	1.29	1.118	0.884	63.66	24.60	1.476	1.122

inference of the MoE teacher (Qwen3-30B-A3B), which activates only 3B parameters per token, while the dense teacher (Qwen3-32B) runs all 32B.

3.4 Broader evaluation and deployment

Broader benchmarks. Since fixed experts could lose domain-specific capability, we evaluate the four extended-training checkpoints on code, math, instruction following, long context, multilingual, and agentic tasks (Table 5). As these are base models, we rely primarily on bits-per-byte (BPB), the prefill-only negative log-likelihood of held-out gold responses normalized by UTF-8 bytes, alongside few-shot and zero-shot accuracy (setup and further science benchmarks in Appendix Q). DO-ACP leads on seven of the nine measures; D2D is marginally lower on HumanEval and multilingual BPB. The advantage over SF is clearer than on the core suite: $+9.1$ pp on RULER, $+8.8$ pp on Belebele, and $5\times$ on GSM8K, versus $+4.6$ pp on the core average.

Calibration sensitivity. DO-ACP selection is robust to calibration size and sequence length (at least 91.7% overlap for sets $4\times$ smaller to $2\times$ larger and for halved or doubled sequence length), but moves by 22–30% across calibration domains. Re-running the pipeline with math or benchmark-derived calibration changes final accuracy by only $+1.0$ to $+1.6$ pp (Appendix R).

Deployment. Relative to its MoE parent, the student has $9.14\times$ fewer parameters and uses $3.0\times$ less peak VRAM, with $1.46\times$ higher throughput and $1.75\times$ faster decode at batch 32×4096 (Appendix S). Compared with the D2D student, which has 3% more parameters, it uses 12.1 GiB less peak VRAM and has 31% lower decode latency, a consequence of its narrower KV cache (4 vs. 8 key-value heads).

3.5 Cross-model validation

To test whether our findings generalize beyond Qwen3, we apply the pipeline to two additional MoE architectures that span different expert counts (32–128), routing widths (4–8), and training stages: **DeepSeek-V2-Lite** [10] (16B, 64 routed + 2 shared experts, top-6, base) and **GPT-OSS-20B** [1] (21B, 32 experts, top-4, post-trained). For each, the dense FFN width matches the teacher's active FFN computation per token; model-specific architectural adjustments are described in Appendix T, and full per-benchmark tables in Appendix U. Figure 6 summarizes the results.

The best configuration uses pure pruning on all architectures. On all three models, the single best configuration uses pure pruning ($K{=}k$): DO-ACP at $K{=}8$ on Qwen3 (Section 3.2), DO-ACP at $K{=}6$ on DeepSeek (42.39% vs. 41.07% at $K{=}12$), and DO-ACP at $K{=}4$ on GPT-OSS (33.71% vs. 32.11% at $K{=}8$). However, the interaction between scoring and K is more nuanced: on DeepSeek, merging ($K{=}12$) outperforms pure pruning ($K{=}6$) for SF ($+1.1$ pp), CP ($+2.5$ pp), and ACP ($+0.6$ pp), while only DO-ACP benefits from pure pruning ($+1.3$ pp). This suggests that diversity-aware scoring selects experts that are individually strong enough to stand alone, whereas frequency-based methods select redundant generalists that benefit from averaging.

Selection benefit is smaller on models with smaller expert pools. DO-ACP and ACP remain among the top scoring methods on both additional models, but the advantage of diversity-aware scoring diminishes with smaller expert pools. The scoring gap (best vs. worst method at $K{=}k$) shrinks from 7.1 pp on Qwen3 (128 experts) to 4.3 pp on DeepSeek (64 experts) to 1.6 pp on

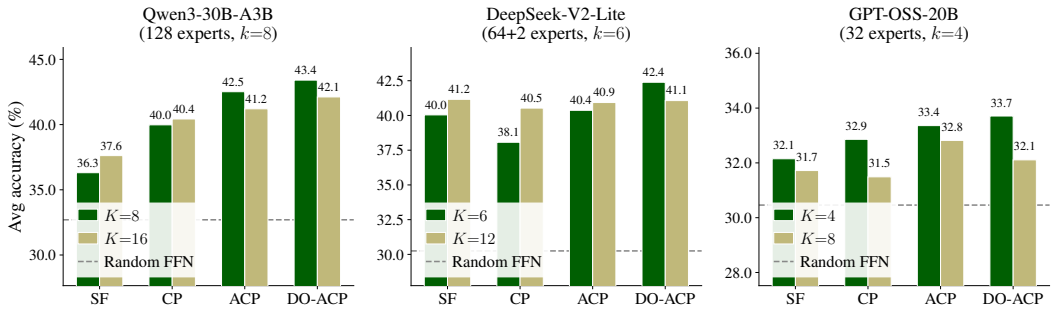


Figure 6: Cross-model validation: post-distillation accuracy (0.3B tokens) by scoring method and K on three MoE architectures. Green bars: pure pruning (K=k, one expert per slot), coral bars: merging (K>k, multiple experts averaged per slot). Dashed line: random FFN baseline. The scoring hierarchy and the benefit of pure pruning both persist across architectures, though the gap compresses with smaller expert pools.

GPT-OSS (32 experts). The overall benefit of MoE-to-dense over random baselines follows the same trend: +13.3 pp, +12.1 pp, and +3.7 pp respectively. We report this as an observed cross-model trend rather than a causal conclusion, since GPT-OSS also differs in its post-training and native reasoning format. One plausible explanation is redundancy in the expert pool: with 128 experts and K=8, many candidates are interchangeable, giving diversity-aware methods room to outperform naive selection, while with only 32 experts and K=4, each expert handles a larger share of tokens, leaving less room for scoring to differentiate. On Qwen3, post-training alone shifts the initialization advantage by only 0.8 pp (Appendix L), which is consistent with, but does not isolate, an effect of pool size.

Appendix V examines compatibility with modularity-aware pretraining [41].

4 Conclusion

We presented the first systematic framework for converting a Mixture-of-Experts language model into a fully dense architecture via expert scoring, grouping, and concatenating, followed by knowledge distillation. We introduced a diversity-aware expert selection criterion (DO-ACP) based on the Gram log-determinant that jointly maximizes expert importance and mutual diversity. We evaluated 350 scoring×grouping×scaling×K combinations on Qwen3-30B-A3B (128 experts) and validated findings on DeepSeek-V2-Lite (64+2 experts) and GPT-OSS-20B (32 experts).

Our evaluation reveals that expert scoring is the dominant design axis (5.7 pp spread vs. ∼1 pp for grouping), with DO-ACP achieving the best accuracy across all configurations and all three models. Diversity-aware scoring enables effective pure pruning: on every architecture, the best configuration retains exactly K=k experts with no weight averaging. MoE-to-dense outperforms dense-to-dense pruning by +6.3 pp after ∼4B-token distillation at 1.6× faster training wall-clock speed. Together, these findings point to a simple recipe: scoring with DO-ACP, retaining exactly the top-k experts per layer, and distilling with forward KL.

5 Limitations

When merging is used ($K > k$), our framework ties merge weights to selection scores, and decoupling these may improve such configurations. Our extended training reaches ∼4B tokens; scaling to tens of billions is needed to establish the quality ceiling, and at this budget converted models are not yet competitive with off-the-shelf dense models, which are reference points rather than controlled baselines: our goal is to derive a dense model from an already-trained MoE more efficiently than training from scratch. The +6.3 pp comparison uses one-shot D2D, as iterative D2D was evaluated only at 0.3B tokens. Fixed experts may lose domain-specific capabilities, and the method does not aim to preserve a post-trained teacher's native behavior. The smaller benefit on GPT-OSS than on Qwen3 does not isolate expert pool size, since GPT-OSS is also a post-trained reasoning model evaluated in completion mode.

Acknowledgments and Disclosure of Funding

This work was conducted at KRAFTON as part of the authors' research and received no third-party funding. Junhyuck Kim, Jihun Yun, Gyeongman Kim, Joonghyun Bae, and Jaewoong Cho are employees of KRAFTON. The authors declare no other competing interests.

References

[1] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[2] J. Austin, A. Odena, M. Nye, M. Bosma, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[3] S. Bai, H. Li, J. Zhang, Z. Hong, and S. Guo. DiEP: Adaptive mixture-of-experts compression through differentiable expert pruning. *arXiv preprint arXiv:2509.16105*, 2025.

[4] L. Bandarkar, D. Liang, B. Muller, M. Artetxe, S. N. Shukla, D. Husa, N. Goyal, A. Krishnan, L. Zettlemoyer, and M. Khabsa. The Belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[5] I.-C. Chen, H.-S. Liu, W.-F. Sun, C.-H. Chao, Y.-C. Hsu, and C.-Y. Lee. Retraining-free merging of sparse MoE via hierarchical clustering. In *International Conference on Machine Learning*, 2025.

[6] M. Chen, J. Tworek, H. Jun, Q. Yuan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[7] T. Chen, S. Huang, Y. Xie, B. Jiao, D. Jiang, H. Zhou, J. Li, and F. Wei. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*, 2022.

[8] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. Li, P. Huang, F. Luo, C. Ruan, Z. Sui, and W. Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[10] DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[11] DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. Technical report, 2026. URL `https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf`.

[12] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39, 2022.

[13] Google DeepMind. Gemma 4. `https://deepmind.google/models/gemma/gemma-4/`, 2026.

[14] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*, 2021.

[15] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, et al. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024.

[18] S. Jha, M. Hashemzadeh, A. Saheb Pasand, A. Parviz, M.-J. Lee, and B. Knyazev. REAM: Merging improves pruning of experts in LLMs. *arXiv preprint arXiv:2604.04356*, 2026.

[19] B.-K. Kim, G. Kim, T.-H. Kim, T. Castells, S. Choi, J. Shin, and H.-K. Song. Shortened llama: Depth pruning for large language models with comparison of retraining methods. *arXiv preprint arXiv:2402.02834*, 2024.

[20] G. Kim, G. Chu, and E. Yang. Every expert matters: Towards effective knowledge distillation for mixture-of-experts language models. *arXiv preprint arXiv:2502.12947*, 2025.

[21] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with PagedAttention. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2023.

[22] M. Lasby, I. Lazarevich, N. Sinnadurai, S. Lie, Y. Ioannou, and V. Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. *arXiv preprint arXiv:2510.13999*, 2025.

[23] L. Li, Z. Qiyuan, J. Wang, W. Li, H. Gu, S. Han, and Y. Guo. Sub-MoE: Efficient mixture-of-expert LLMs compression via subspace expert merging. *arXiv preprint arXiv:2506.23266*, 2025.

[24] P. Li, Z. Zhang, P. Yadav, Y.-L. Sung, Y. Cheng, M. Bansal, and T. Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy. In *International Conference on Learning Representations*, 2024.

[25] Z. Li, C. Liang, Z. Zhang, I. Hong, Y. J. Kim, W. Chen, and T. Zhao. SlimMoE: Structured compression of large MoE models via expert slimming and distillation. *arXiv preprint arXiv:2506.18349*, 2025.

[26] A. H. Liu, K. Khandelwal, S. Subramanian, V. Jouault, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

[27] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2017.

[28] Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`, 2025.

[29] R. Miao, Y. Yao, Z. Wang, Z. Wang, B. Yi, L. Liu, Y. Zhao, and T. Yang. MergeMoE: Efficient compression of MoE models via expert output merging. *arXiv preprint arXiv:2510.14436*, 2025.

[30] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[31] S. Muralidharan, S. T. Sreenivas, R. Joshi, M. Chochowski, M. Patwary, M. Shoeybi, B. Catanzaro, J. Kautz, and P. Molchanov. Compact language models via pruning and knowledge distillation. *arXiv preprint arXiv:2407.14679*, 2024.

[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[33] D. V. Nguyen, A. T. Nguyen, M. H. Nguyen, L. Q. Nguyen, S. Jiang, E. Fetaya, L. D. Tran, G. Chechik, and T. M. Nguyen. Expert merging in sparse mixture of experts with nash bargaining. *arXiv preprint arXiv:2510.16138*, 2025.

[34] G. Penedo, H. Kydlíček, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, T. Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

[35] F. Pukelsheim. *Optimal Design of Experiments*. SIAM, 2006.

[36] O. Roy and M. Vetterli. The effective rank: A measure of effective dimensionality. *15th European Signal Processing Conference*, pages 606–610, 2007.

[37] N. Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[38] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[39] S. T. Sreenivas, S. Muralidharan, R. Joshi, M. Chochowski, M. Patwary, M. Shoeybi, B. Catanzaro, J. Kautz, and P. Molchanov. LLM pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*, 2024.

[40] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[41] R. Wang, A. Bhagia, and S. Min. EMO: Pretraining mixture of experts for emergent modularity. *arXiv preprint arXiv:2605.06663*, 2026.

[42] J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text (W-NUT)*, 2017.

[43] M. Xia, T. Gao, Z. Zeng, and D. Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2024.

[44] Y. Xie, Z. Zhang, D. Zhou, C. Xie, Z. Song, X. Liu, Y. Wang, X. Lin, and A. Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*, 2024.

[45] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[46] C. Yang, Y. Sui, J. Xiao, L. Huang, Y. Gong, Y. Duan, W. Jia, M. Yin, Y. Cheng, and B. Yuan. MoE-I^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of EMNLP*, 2024.

[47] A. Zeng, X. Lv, Z. Hou, Z. Du, Q. Zheng, B. Chen, D. Yin, C. Ge, C. Huang, C. Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[48] Y. Zhao, Z. Wang, and M. Zhang. PuzzleMoE: Efficient compression of large mixture-of-experts models via sparse expert merging and bit-packed inference. *arXiv preprint arXiv:2511.04805*, 2025.

A Related work

MoE expert pruning. A growing line of work reduces the number of active experts while retaining sparse routing. REAP [22] scores experts by gate value $\times$ activation norm and removes the lowest-scoring ones. We adapt this principle into a factorized variant and combine it with the D-optimal selection criterion (Section 2.2). SlimMoE [25] retains all experts but progressively prunes neurons within each expert through multi-stage distillation. MoE-Pruner [44] one-shot prunes weights within experts using a scoring rule that combines weight magnitude, input activation, and router weight. DiEP [3] learns non-uniform pruning rates per layer via differentiable optimization. MoE-I^2 [46] scores experts by per-expert loss degradation, requiring $E \times L$ forward passes. Earlier task-specific MoE pruning [7] selects experts specialized for a target downstream task. EMO [41] modifies MoE pretraining to encourage emergent expert modularity, enabling subset pruning at deployment time. All these methods produce *smaller MoE* models, whereas our work produces a fully dense model.

MoE expert merging. Rather than removing experts, merging methods combine multiple experts into fewer groups. REAM [18] shows that merging experts by router-weighted averaging can outperform pruning them entirely. MC-SMoE [24] groups experts around most frequently activated experts using router-logit similarity, then merges each group via permutation-aligned frequency-weighted averaging. HC-SMoE [5] showed that output-similarity clustering outperforms router-logit and weight similarity for grouping. Sub-MoE [23] uses output-based K-means++ clustering before joint SVD merging. MergeMoE [29] proves the optimality of frequency-based weighting and clusters by concatenated gate/up weight similarity. PuzzleMoE [48] constructs new experts via entry-wise selection that combines weight similarity and activation-weight saliency. NAMEx [33] computes merging coefficients via Nash Bargaining optimization. These methods all produce smaller MoE or specialized sparse models, while our pipeline uses expert selection and merging as an initialization for a dense model refined by knowledge distillation.

Dense model compression and distillation. Minitron [31, 39] showed that activation-based importance scoring followed by width/depth pruning and logit-level knowledge distillation produces compact LLMs competitive with models trained from scratch. Other approaches include targeted depth removal [19] and joint structured pruning with continued pre-training [43]. Our pipeline follows the general flow of pruning first and then recovering with knowledge distillation, but the pruning focuses on selecting a smaller set of experts and converting to a dense model, and the teacher model stays MoE during distillation.

Subset selection and D-optimal design. Selecting a diverse, high-quality subset from a large candidate pool is a classical problem in experimental design [35]. The D-optimality criterion maximizes the log-determinant of an information matrix and is monotone submodular, admitting greedy $(1-1/e)$-approximation [32]. Our D-Optimal expert selection (Section 2.2) instantiates this in the expert output space: the importance-weighted Gram matrix captures both individual quality and pairwise redundancy, so maximizing its log-determinant selects experts that are jointly informative rather than individually top-ranked. To our knowledge, this is the first application of D-Optimal subset selection to MoE compression.

B Block concatenation preserves representative activations

We show that, once a layer has been converted into k representative FFNs, block concatenation produces the same intermediate activations as those representatives and differs only in the final static aggregation weights. When $K{=}k$, the representatives are copied experts, so this is an exact statement about the selected original experts. When $K{>}k$, the representatives are first formed by parameter-space averaging; the result below applies to those constructed representatives, not to every original expert before averaging.

Consider a single representative FFN with SwiGLU activation. Representative g computes:

$$f_g(\mathbf{h}) = \mathbf{W}^{(g)}_{\text{down}}\big(\sigma(\mathbf{W}^{(g)}_{\text{gate}}\mathbf{h}) \odot \mathbf{W}^{(g)}_{\text{up}}\mathbf{h}\big), \tag{5}$$

where σ is the SiLU activation and $\odot$ is element-wise multiplication. The original MoE output uses token-dependent routing weights:

$$f_{\text{MoE}}(\mathbf{h}) = \sum_{e \in \mathcal{S}(\mathbf{h})} w_e(\mathbf{h}) \cdot f_e(\mathbf{h}), \tag{6}$$

where $\mathcal{S}(\mathbf{h})$ is the top-k routed expert set and $w_e(\mathbf{h})$ are the router weights.

Now consider the concatenated dense FFN with $\mathbf{W}_{\text{gate}} = [\mathbf{W}_{\text{gate}}^{(1)}; \ldots; \mathbf{W}_{\text{gate}}^{(k)}]$ and similarly for $\mathbf{W}_{\text{up}}$. Since row-concatenation distributes over matrix–vector multiplication:

$$\mathbf{W}_{\text{gate}}\mathbf{h} = \begin{bmatrix} \mathbf{W}_{\text{gate}}^{(1)}\mathbf{h} \\ \vdots \\ \mathbf{W}_{\text{gate}}^{(k)}\mathbf{h} \end{bmatrix}, \qquad \mathbf{W}_{\text{up}}\mathbf{h} = \begin{bmatrix} \mathbf{W}_{\text{up}}^{(1)}\mathbf{h} \\ \vdots \\ \mathbf{W}_{\text{up}}^{(k)}\mathbf{h} \end{bmatrix}. \tag{7}$$

The SwiGLU activation applies element-wise, so the intermediate activation decomposes into k independent blocks, each identical to the corresponding representative's intermediate activation. The column-concatenated down-projection then sums over these blocks:

$$f_{\text{dense}}(\mathbf{h}) = \mathbf{W}_{\text{down}}\big(\sigma(\mathbf{W}_{\text{gate}}\mathbf{h}) \odot \mathbf{W}_{\text{up}}\mathbf{h}\big) = \sum_{g=1}^{k} \alpha_g \cdot f_g(\mathbf{h}). \tag{8}$$

The representative intermediate activations (gate projections, up projections, and SwiGLU outputs) are *exactly* preserved by block concatenation. Within this block-concatenation step, the only approximation is that the dense model uses static α_g from down-projection scaling (Section 2.4) instead of token-dependent router weights; the full MoE-to-dense conversion also fixes the selected experts and, when $K > k$, replaces groups of original experts by parameter-averaged representatives. Knowledge distillation compensates for these approximations.

C Algorithms

This section presents the two core procedures of our pipeline: the per-layer MoE-to-dense conversion (Algorithm 1) and the greedy D-Optimal subset selection used by the DO scoring family (Algorithm 2).

MoE-to-dense conversion. Algorithm 1 gives the per-layer conversion, where E is the number of experts, k the MoE top-k count (also the number of groups), $K \geq k$ the experts selected for conversion, $\mathcal{G}_g \subseteq [E]$ the experts assigned to group $g \in [k]$, and α_g the down-projection scaling factor (uniform $1/k$ or score-proportional, Section 2.4).

Algorithm 1 MoE-to-dense conversion for layer ℓ

Require: Experts $\{\mathbf{W}^{(e)}\}_{e=1}^{E}$, scores $\{s^{(e)}\}_{e=1}^{E}$, top-K, number of groups k; scoring, grouping, and scaling method choices.
Ensure: Dense projections $\mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{up}} \in \mathbb{R}^{d_{\text{dense}} \times d}$, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times d_{\text{dense}}}$

Score: Compute scores $s^{(e)}$ for all E experts; select top-K.
Group: Partition the K selected experts into k groups using the chosen grouping method.
for each group g **do**
  Compute within-group weights: $w^{(e)} = s^{(e)} \big/ \sum_{e' \in \mathcal{G}_g} s^{(e')}$ for each $e \in \mathcal{G}_g$.
  for proj $\in \{\text{gate}, \text{up}, \text{down}\}$ **do**
    Merge: $\mathbf{W}_{\text{proj}}^{(g)} = \sum_{e \in \mathcal{G}_g} w^{(e)}\, \mathbf{W}_{\text{proj}}^{(e)}$ ▷ Weighted average (identity for $|\mathcal{G}_g| = 1$)
  end for
  Scale: $\tilde{\mathbf{W}}_{\text{down}}^{(g)} = \alpha_g \cdot \mathbf{W}_{\text{down}}^{(g)}$ ▷ Uniform or proportional
end for
Concatenate: $\mathbf{W}_{\text{gate}} = [\mathbf{W}_{\text{gate}}^{(1)}; \ldots; \mathbf{W}_{\text{gate}}^{(k)}]$, $\mathbf{W}_{\text{up}} = [\mathbf{W}_{\text{up}}^{(1)}; \ldots; \mathbf{W}_{\text{up}}^{(k)}]$ ▷ Row-concat
  $\mathbf{W}_{\text{down}} = [\tilde{\mathbf{W}}_{\text{down}}^{(1)}, \ldots, \tilde{\mathbf{W}}_{\text{down}}^{(k)}]$ ▷ Column-concat

The algorithm applies independently to each of the L MoE layers. All non-MoE parameters (attention, embeddings, layer norms) are copied unchanged from teacher to student.

Greedy D-Optimal selection. Algorithm 2 gives the greedy size-K subset selection used by the DO scoring family, instantiating the $(1-1/e)$-approximation of Proposition F.1. The importance-weighted Gram matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{E\times E}$ has entries $\boldsymbol{\mathcal{K}}_{ij} = \sqrt{I_i I_j}\,\mathbf{G}_{ij}$, with base importance $I_e = s^{(e)} \geq 0$ (CP or ACP) and Gram entries $\mathbf{G}_{ij} = \mathbb{E}_t[\langle f_i(t), f_j(t)\rangle]$ over calibration tokens. For $S \subseteq [E]$, $\boldsymbol{\mathcal{K}}_S$ is the principal submatrix indexed by S, and $\boldsymbol{\mathcal{K}}_{eS}$, $\boldsymbol{\mathcal{K}}_{Se}$ the corresponding row and column slices for $e \notin S$. The objective is $F(S) := \log\det(\boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\mathbf{I})$ with $\lambda_{\text{reg}} = \frac{1}{KE}\sum_{e=1}^{E}\boldsymbol{\mathcal{K}}_{ee}$.

Algorithm 2 Greedy D-Optimal expert selection (per layer)

Require: Importance-weighted Gram $\boldsymbol{\mathcal{K}}$ (positive semidefinite), target subset size K, regularizer λ_{reg}.
Ensure: Ordered subset $S \subseteq [E]$ with $|S| = K$

$S \leftarrow \emptyset$
for step $= 1, \ldots, K$ **do**
  if $S = \emptyset$ **then**
    $\text{gain}(e) \leftarrow \log(\boldsymbol{\mathcal{K}}_{ee} + \lambda_{\text{reg}})$ for each $e \in [E]$
  else
    $A_S \leftarrow \boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\mathbf{I}$; precompute A_S^{-1} ▷ once per step
    for each $e \notin S$ **do**
      $\sigma_e \leftarrow \boldsymbol{\mathcal{K}}_{ee} + \lambda_{\text{reg}} - \boldsymbol{\mathcal{K}}_{eS}\,A_S^{-1}\,\boldsymbol{\mathcal{K}}_{Se}$ ▷ Schur complement
      $\text{gain}(e) \leftarrow \log\sigma_e$
    end for
  end if
  $e^\star \leftarrow \arg\max_{e\notin S}\text{gain}(e)$
  $S \leftarrow S \cup \{e^\star\}$
end for
return S

A_S^{-1} is recomputed each step at cost $O(K^3)$, and the Schur complement is evaluated for each of the E candidates at $O(K^2)$ per evaluation, yielding overall $O(K^3E)$ time.

D Scoring baseline definitions

For completeness, we provide the formal definitions of the three routing-based baseline scoring methods (Section 2.2). Let $p_\ell^{(e)}(t)$ be the softmax routing probability and $\mathcal{S}_\ell(t)$ the set of top-k selected experts for token t in layer ℓ.

Selection frequency (SF). $s_\ell^{(e)} = \frac{1}{N}\sum_{t=1}^{N}\mathbb{1}[e \in \mathcal{S}_\ell(t)]$.

Pre-selection probability (PP). $s_\ell^{(e)} = \frac{1}{N}\sum_{t=1}^{N} p_\ell^{(e)}(t)$.

Post-selection probability (PS). $s_\ell^{(e)} = \frac{1}{N}\sum_{t:e\in\mathcal{S}_\ell(t)} p_\ell^{(e)}(t)$.

PS decomposes as SF × CP: $s_\ell^{(e)} = f_\ell^{(e)} \cdot \mathbb{E}[p_\ell^{(e)}(t) \mid e \in \mathcal{S}_\ell(t)]$.

E Excluded scoring and grouping methods

Scoring methods excluded. We do not evaluate Fisher-weighted merging, which MC-SMoE [24] found to underperform frequency weighting (their Table 8). We omit loss-degradation scoring from MoE-I^2 [46], requiring $E \times L = 6{,}144$ forward passes, impractical at our scale. We do not evaluate Nash Bargaining merging coefficients (NAMEx [33]), since our framework ties merge weights to selection scores (Section 5). Per-weight saliency metrics (Wanda [40]) operate at intra-expert granularity, orthogonal to our per-expert framework.

Grouping methods excluded. We do not evaluate K-means clustering, which HC-SMoE [5] found to underperform hierarchical clustering for expert grouping. We omit the element-wise similarity

and saliency masks of PuzzleMoE [48], which merge experts entry by entry and rely on a bit-packed format with a custom decode kernel for efficient inference, rather than selecting whole experts.

F Proofs for D-Optimal expert selection

This section provides full statements and proofs for the theoretical results introduced in Section 2.2. We use the notation established there: $f_e(t) \in \mathbb{R}^d$ is the output of expert e on token t, $I_e = s^{(e)} \geq 0$ is a base importance score (CP or ACP), and $\mathcal{K}_{ij} = \sqrt{I_i I_j} \cdot \mathbb{E}_t[\langle f_i(t), f_j(t) \rangle]$ is the importance-weighted kernel. For any subset $S \subseteq [E]$, we define:

$$F(S) := \log\det\left(\boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\mathbf{I}\right), \qquad \widetilde{F}(S) := \log\det\left(\mathbf{I} + \lambda_{\text{reg}}^{-1}\boldsymbol{\mathcal{K}}_S\right). \tag{9}$$

For fixed cardinality $|S| = K$, F and $\widetilde{F}$ differ by the additive constant $K \log \lambda_{\text{reg}}$. We use $\widetilde{F}$ for the submodularity argument (it is monotone for any $\lambda_{\text{reg}} > 0$, while F is not) and F in the incoherence and stability proofs (it factors cleanly with the diagonal proxy G).

F.1 Redundancy counterexample (Proof of Theorem 2.1)

This counterexample uses a valid positive regularization chosen for the construction. It is intended to show that log-determinant selection can avoid a redundancy failure mode of independent ACP ranking; the empirical algorithm uses the fixed regularization in Section 2.2.

Proof. Fix $K \geq 2$ and set $E = 2K - 1$. Let the calibration space be

$$\mathcal{X} := \{a_1, \ldots, a_K, b_2, \ldots, b_K\}, \tag{10}$$

and let $\mathcal{D}$ be the uniform distribution on $\mathcal{X}$. We construct a top-1 MoE layer with scalar expert outputs. Define

$$f_e(x) = \mathbf{1}\{x \in \{a_1, \ldots, a_K\}\}, \quad e = 1, \ldots, K, \tag{11}$$

$$f_{K+j-1}(x) = \mathbf{1}\{x = b_j\}, \quad j = 2, \ldots, K. \tag{12}$$

Thus the first K experts are identical, while the remaining $K - 1$ experts are pairwise orthogonal specialists.

Define a deterministic router $r : \mathcal{X} \to [E]$ by

$$r(a_j) = j, \quad j = 1, \ldots, K, \tag{13}$$

$$r(b_j) = K + j - 1, \quad j = 2, \ldots, K, \tag{14}$$

and let the routing probabilities be $p^{(e)}(x) := \mathbf{1}\{e = r(x)\}$. The resulting MoE teacher output is

$$F_{\text{MoE}}(x) := \sum_{e=1}^{E} p^{(e)}(x) f_e(x) = 1 \qquad \text{for every } x \in \mathcal{X}. \tag{15}$$

We take ACP as the base importance score. Because the router is deterministic, every selected expert has conditional probability 1, so the ACP score of expert e reduces to $\sqrt{\mathbb{E}[f_e(x)^2]}$. Therefore

$$I_1 = \cdots = I_K =: I_A = \sqrt{\frac{K}{2K-1}}, \qquad I_{K+1} = \cdots = I_{2K-1} =: I_B = \sqrt{\frac{1}{2K-1}}, \tag{16}$$

and hence $I_A > I_B > 0$. Independent top-K ranking by the base importances therefore selects

$$S_{\text{ind}} = \{1, \ldots, K\}. \tag{17}$$

For any subset $S \subseteq [E]$, define its best linear reconstruction error against the teacher output by

$$\mathcal{L}(S) := \inf_{a \in \mathbb{R}^{|S|}} \mathbb{E}_{x \sim \mathcal{D}} \left[\left(\sum_{e \in S} a_e f_e(x) - F_{\text{MoE}}(x) \right)^2 \right]. \tag{18}$$

Independent ranking incurs constant error. For S_{ind}, every selected feature equals $\mathbf{1}\{x \in \{a_1, \ldots, a_K\}\}$. Hence every linear combination over S_{ind} has the form

$$x \mapsto c\,\mathbf{1}\{x \in \{a_1, \ldots, a_K\}\} \tag{19}$$

for some scalar c. The choice $c = 1$ is optimal, because it matches $F_{\text{MoE}}(x) = 1$ on the K points $a_1, \ldots, a_K$. On the remaining $K - 1$ points $b_2, \ldots, b_K$, the reconstruction is zero, so the squared error is one. Therefore

$$\mathcal{L}(S_{\text{ind}}) = \frac{K-1}{2K-1}. \tag{20}$$

Log-det selects a zero-error subset. Now define

$$S_{\text{good}} := \{1\} \cup \{K+1, \ldots, 2K-1\}. \tag{21}$$

For this subset, choosing coefficient vector $a \equiv 1$ gives

$$\sum_{e \in S_{\text{good}}} a_e f_e(x) = \mathbf{1}\{x \in \{a_1, \ldots, a_K\}\} + \sum_{j=2}^{K} \mathbf{1}\{x = b_j\} = 1 = F_{\text{MoE}}(x) \tag{22}$$

for every $x \in \mathcal{X}$, so $\mathcal{L}(S_{\text{good}}) = 0$.

To identify the log-determinant maximizer, let

$$\alpha := \mathcal{K}_{11} = I_A\,\mathbb{E}[f_1(x)^2] = \left(\frac{K}{2K-1}\right)^{3/2}, \qquad \beta := \mathcal{K}_{K+1,K+1} = I_B\,\mathbb{E}[f_{K+1}(x)^2] = \left(\frac{1}{2K-1}\right)^{3/2}. \tag{23}$$

Because the first block of experts is identical and the specialist block is orthogonal to it and to itself off-diagonal, every size-K subset S is characterized by

$$t := |S \cap \{1, \ldots, K\}| \in \{1, \ldots, K\}, \tag{24}$$

and the nonzero eigenvalues of $\boldsymbol{\mathcal{K}}_S$ are

$$t\alpha \qquad \text{and} \qquad \underbrace{\beta, \ldots, \beta}_{K-t \text{ times}}. \tag{25}$$

The remaining $t - 1$ eigenvalues are zero. For this construction, setting $\lambda_{\text{reg}} := \beta$, we obtain

$$\det(\boldsymbol{\mathcal{K}}_S + \lambda_{\text{reg}}\mathbf{I}) = (\beta + t\alpha)\,\beta^{\,t-1}(2\beta)^{K-t} = \beta^{K-1}2^{K-t}(\beta + t\alpha). \tag{26}$$

Let $D(t) := \beta^{K-1}2^{K-t}(\beta + t\alpha)$. For every $t \geq 1$,

$$\frac{D(t+1)}{D(t)} = \frac{\beta + (t+1)\alpha}{2(\beta + t\alpha)} < 1 \tag{27}$$

because

$$\beta + (t+1)\alpha < 2\beta + 2t\alpha \iff (1-t)\alpha < \beta, \tag{28}$$

and the right-hand side is true for all $t \geq 1$ since $\beta > 0$. Thus $D(t)$ is strictly decreasing in t, so every maximizer of the log-determinant objective has $t = 1$. Every such maximizer has the form

$$\{i\} \cup \{K+1, \ldots, 2K-1\} \qquad \text{for some } i \in \{1, \ldots, K\}. \tag{29}$$

Because all first-block experts are identical, each of these maximizing subsets has the same reconstruction error as S_{good}, namely zero. Consequently, any size-K subset maximizing the log-determinant objective achieves zero reconstruction error. □

F.2 Submodularity and greedy guarantee (Proposition F.1)

Proposition F.1 (Greedy D-Optimal selection is near-optimal)**.** *For any positive semidefinite kernel $\mathcal{K}$ and any $\lambda_{\text{reg}} > 0$, the normalized objective $\widetilde{F}(S) = \log\det(\mathbf{I} + \lambda_{\text{reg}}^{-1}\mathcal{K}_S)$ is monotone submodular. The greedy algorithm returns S_{greedy} with $\widetilde{F}(S_{\text{greedy}}) \geq (1 - 1/e)\max_{|S|=K} \widetilde{F}(S)$.*

D-Optimal selection is therefore an instance of cardinality-constrained D-optimal design in expert-output space. The result holds for any base importance I_e, and we instantiate it with both CP and ACP (yielding DO-CP and DO-ACP).

Proof. Fix $S \subseteq [E]$ and $e \notin S$. Let $A_S := \mathcal{K}_S + \lambda_{\text{reg}}\mathbf{I}$ and define the Schur complement $\sigma_e(S) := \mathcal{K}_{ee} + \lambda_{\text{reg}} - \mathcal{K}_{eS}A_S^{-1}\mathcal{K}_{Se}$. By the block-determinant formula:

$$\det\big(\mathcal{K}_{S\cup\{e\}} + \lambda_{\text{reg}}\mathbf{I}\big) = \det(A_S)\cdot\sigma_e(S). \tag{30}$$

Now

$$\begin{bmatrix} A_S & \mathcal{K}_{Se} \\ \mathcal{K}_{eS} & \mathcal{K}_{ee} \end{bmatrix} = \begin{bmatrix} \mathcal{K}_S & \mathcal{K}_{Se} \\ \mathcal{K}_{eS} & \mathcal{K}_{ee} \end{bmatrix} + \begin{bmatrix} \lambda_{\text{reg}}\mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \succeq 0, \tag{31}$$

because both summands are positive semidefinite and $A_S \succ 0$. Taking the Schur complement with respect to the positive-definite block A_S gives

$$\mathcal{K}_{ee} - \mathcal{K}_{eS}A_S^{-1}\mathcal{K}_{Se} \geq 0. \tag{32}$$

Therefore $\sigma_e(S) \geq \lambda_{\text{reg}}$, and hence

$$\widetilde{F}(S\cup\{e\}) - \widetilde{F}(S) = \log\left(\frac{\sigma_e(S)}{\lambda_{\text{reg}}}\right) \geq 0, \tag{33}$$

which proves monotonicity.

For diminishing returns, fix $S \subseteq T \subseteq [E]$ and $e \notin T$. We show $\sigma_e(S) \geq \sigma_e(T)$, i.e., $\mathcal{K}_{eT}A_T^{-1}\mathcal{K}_{Te} \geq \mathcal{K}_{eS}A_S^{-1}\mathcal{K}_{Se}$. By the variational characterization of quadratic forms with positive-definite matrices:

$$\mathcal{K}_{eT}A_T^{-1}\mathcal{K}_{Te} = \max_{z\in\mathbb{R}^{|T|}} \big\{2z^\top\mathcal{K}_{Te} - z^\top A_T z\big\}. \tag{34}$$

Restricting z to have support only on S (i.e., $z = (z_S, \mathbf{0}_{T\setminus S})$), the $S \times S$ principal subblock of A_T is A_S, so the restricted maximum equals $\mathcal{K}_{eS}A_S^{-1}\mathcal{K}_{Se}$. Since the unrestricted maximum is at least the restricted one:

$$\mathcal{K}_{eT}A_T^{-1}\mathcal{K}_{Te} \geq \mathcal{K}_{eS}A_S^{-1}\mathcal{K}_{Se}. \tag{35}$$

Hence $\widetilde{F}(S\cup\{e\}) - \widetilde{F}(S) = \log(\sigma_e(S)/\lambda_{\text{reg}}) \geq \log(\sigma_e(T)/\lambda_{\text{reg}}) = \widetilde{F}(T\cup\{e\}) - \widetilde{F}(T)$, which is submodularity. The $(1-1/e)$ greedy bound follows from Nemhauser et al. [32]. □

F.3 Incoherence bound (Theorem F.3)

Definition F.2 (Mutual coherence)**.** *For $i \neq j$ with $\mathcal{K}_{ii}\mathcal{K}_{jj} > 0$, define $\rho_{ij} := \mathcal{K}_{ij}/\sqrt{\mathcal{K}_{ii}\mathcal{K}_{jj}}$ and $\mu := \max_{i\neq j}|\rho_{ij}|$.*

Theorem F.3 (When does D-Optimal selection help the most?)**.** *Let $G(S) := \sum_{e\in S}\log(\mathcal{K}_{ee} + \lambda_{\text{reg}})$ and $S_{\text{diag}} \in \arg\max_{|S|=K} G(S)$. If $(K-1)\mu < 1$, then*

$$F(S_{\text{diag}}) \geq \max_{|S|=K} F(S) - K\log\left(\frac{1+(K-1)\mu}{1-(K-1)\mu}\right).$$

When experts are nearly incoherent ($\mu \approx 0$), the diagonal proxy is a good approximation to the full log-determinant objective; the off-diagonal diversity interactions contribute less.

Proof. Fix S with $|S| = K$. Define the matrices:

$$B_S := \mathrm{diag}(\mathcal{K}_{ee} + \lambda_{\mathrm{reg}})_{e\in S}, \qquad R_S := B_S^{-1/2}(\boldsymbol{\mathcal{K}}_S - \mathrm{diag}(\mathcal{K}_{ee})_{e\in S})B_S^{-1/2}. \tag{36}$$

Then $\boldsymbol{\mathcal{K}}_S + \lambda_{\mathrm{reg}}\mathbf{I} = B_S^{1/2}(\mathbf{I} + R_S)B_S^{1/2}$, so $F(S) = G(S) + \log\det(\mathbf{I} + R_S)$.

The matrix R_S is symmetric with zero diagonal, and every off-diagonal entry has magnitude at most μ. By Gershgorin's theorem, every eigenvalue of R_S lies in $[-(K-1)\mu,\, (K-1)\mu]$. Since $(K-1)\mu < 1$ by assumption, $\mathbf{I} + R_S$ is positive definite with eigenvalues in $[1-(K-1)\mu,\, 1+(K-1)\mu]$. Taking logarithms and summing over the K eigenvalues:

$$K\log\big(1-(K-1)\mu\big) \le \log\det(\mathbf{I} + R_S) \le K\log\big(1+(K-1)\mu\big). \tag{37}$$

Applying the upper bound to $S^\star = \arg\max_{|S|=K} F(S)$ and the lower bound to $S_{\mathrm{diag}} = \arg\max_{|S|=K} G(S)$, and using $G(S_{\mathrm{diag}}) \ge G(S^\star)$:

$$F(S_{\mathrm{diag}}) \ge F(S^\star) - K\log\left(\frac{1+(K-1)\mu}{1-(K-1)\mu}\right). \tag{38}$$

□

G Additional theoretical results

G.1 Finite-sample calibration guarantee

In practice, the DO kernel is estimated from a finite calibration set using the CP or ACP scores defined in Section 2.2. The following theorem analyzes those estimators directly.

Theorem G.1 (Uniform stability of empirical DO-CP and DO-ACP)**.** *For each token t, let $\mathcal{S}(t) = \text{top-k}\{p^{(e)}(t)\}_{e=1}^{E}$, with any fixed tie-breaking rule, and define*

$$z_e(t) := \mathbf{1}\{e \in \mathcal{S}(t)\}, \tag{39}$$

$$q_e := \mathbb{E}[z_e(t)], \qquad a_e := \mathbb{E}[z_e(t)\, p^{(e)}(t)], \qquad v_e := \mathbb{E}[\|f_e(t)\|_2^2], \tag{40}$$

$$\mathrm{CP}_e := \frac{a_e}{q_e}, \qquad \mathrm{ACP}_e := \frac{a_e}{q_e}\sqrt{v_e}, \qquad G_{ij} := \mathbb{E}[\langle f_i(t), f_j(t)\rangle]. \tag{41}$$

Given i.i.d. calibration tokens $t_1, \dots, t_n \sim \mathcal{D}$, define empirical quantities

$$\widehat{q}_e := \frac{1}{n}\sum_{m=1}^{n} z_e(t_m), \qquad \widehat{a}_e := \frac{1}{n}\sum_{m=1}^{n} z_e(t_m)\, p^{(e)}(t_m), \tag{42}$$

$$\widehat{v}_e := \frac{1}{n}\sum_{m=1}^{n} \|f_e(t_m)\|_2^2, \qquad \widehat{G}_{ij} := \frac{1}{n}\sum_{m=1}^{n} \langle f_i(t_m), f_j(t_m)\rangle. \tag{43}$$

Set

$$\widehat{\mathrm{CP}}_e := \begin{cases} \widehat{a}_e/\widehat{q}_e, & \widehat{q}_e > 0, \\ 0, & \widehat{q}_e = 0, \end{cases} \tag{44}$$

$$\widehat{\mathrm{ACP}}_e := \widehat{\mathrm{CP}}_e\sqrt{\widehat{v}_e}. \tag{45}$$

Choose the base score either as DO-CP, with

$$I_e := \mathrm{CP}_e, \qquad \widehat{I}_e := \widehat{\mathrm{CP}}_e, \tag{46}$$

or as DO-ACP, with

$$I_e := \mathrm{ACP}_e, \qquad \widehat{I}_e := \widehat{\mathrm{ACP}}_e. \tag{47}$$

Let

$$\mathcal{K}_{ij} := \sqrt{I_i I_j}\, G_{ij}, \qquad \widehat{\mathcal{K}}_{ij} := \sqrt{\widehat{I}_i \widehat{I}_j}\, \widehat{G}_{ij}. \tag{48}$$

For a target subset size K, set the population and empirical regularizers by the same diagonal rule as Section 2.2:

$$\lambda_{\mathrm{reg}} := \frac{1}{KE}\sum_{e=1}^{E}\mathcal{K}_{ee}, \qquad \widehat{\lambda}_{\mathrm{reg}} := \frac{1}{KE}\sum_{e=1}^{E}\widehat{\mathcal{K}}_{ee}, \tag{49}$$

Assume $\lambda_{\mathrm{reg}} > 0$, and define

$$F(S) := \log\det(\boldsymbol{\mathcal{K}}_S + \lambda_{\mathrm{reg}}\mathbf{I}), \qquad \widehat{F}(S) := \log\det(\widehat{\boldsymbol{\mathcal{K}}}_S + \widehat{\lambda}_{\mathrm{reg}}\mathbf{I}). \tag{50}$$

Assume $\|f_e(t)\|_2 \le B_f$ almost surely for all $e \in [E]$, $q_e \ge q_{\min} > 0$ for all e, and $I_e \in [I_{\min}, I_{\max}]$ for all e. In the DO-ACP case, also assume $v_e \ge v_{\min} > 0$ for all e. Define

$$r_n := \sqrt{\frac{2\log(8E^2/\delta)}{n}}, \tag{51}$$

$$C_{\mathrm{CP}} := \frac{4}{q_{\min}}, \tag{52}$$

$$C_{\mathrm{ACP}} := \frac{4B_f}{q_{\min}} + \frac{B_f^2}{\sqrt{2v_{\min}}}, \tag{53}$$

$$C_I := \begin{cases} C_{\mathrm{CP}}, & \textit{for DO-CP}, \\ C_{\mathrm{ACP}}, & \textit{for DO-ACP}, \end{cases} \tag{54}$$

$$C_h := B_f^2\sqrt{\frac{2I_{\max}}{I_{\min}} + 1} + I_{\max} + \frac{I_{\min}}{2}, \tag{55}$$

$$\varepsilon_n := C_h \max\{C_I, B_f^2\}\, r_n, \tag{56}$$

$$\Delta_n := \left(K + \frac{1}{K}\right)\varepsilon_n. \tag{57}$$

If

$$r_n \le \frac{q_{\min}}{2}, \qquad C_I r_n \le \frac{I_{\min}}{2}, \tag{58}$$

and, in the DO-ACP case,

$$B_f^2 r_n \le \frac{v_{\min}}{2}, \tag{59}$$

then with probability at least $1-\delta$:

$$\max_{i,j\in[E]} |\widehat{\mathcal{K}}_{ij} - \mathcal{K}_{ij}| \le \varepsilon_n. \tag{60}$$

Consequently, if $\Delta_n < \lambda_{\mathrm{reg}}$, then

$$\left|\widehat{F}(S) - F(S)\right| \le -K\log\left(1 - \frac{\Delta_n}{\lambda_{\mathrm{reg}}}\right) \qquad \textit{for all } |S| = K, \tag{61}$$

and if $\widehat{S}$ maximizes $\widehat{F}$ while $S^\star$ maximizes F, then

$$F(\widehat{S}) \ge F(S^\star) - K\log\left(\frac{1 + \Delta_n/\lambda_{\mathrm{reg}}}{1 - \Delta_n/\lambda_{\mathrm{reg}}}\right). \tag{62}$$

Proof. We proceed in four steps. Write $\lambda := \lambda_{\mathrm{reg}}$ for brevity.

Step 1: Uniform concentration of the empirical moments. For each $e \in [E]$, the random variables $z_e(t)$ and $z_e(t)p^{(e)}(t)$ lie in $[0,1]$, while $\|f_e(t)\|_2^2$ lies in $[0, B_f^2]$. Also, for every $i, j \in [E]$,

$$|\langle f_i(t), f_j(t)\rangle| \le \|f_i(t)\|_2\|f_j(t)\|_2 \le B_f^2 \tag{63}$$

almost surely. Therefore Hoeffding's inequality gives

$$\Pr\left(\max_{e\in[E]}|\widehat{q}_e - q_e| > r_n\right) \le 2E\exp(-2nr_n^2), \tag{64}$$

$$\Pr\left(\max_{e\in[E]}|\widehat{a}_e - a_e| > r_n\right) \le 2E\exp(-2nr_n^2), \tag{65}$$

$$\Pr\left(\max_{e\in[E]}|\widehat{v}_e - v_e| > B_f^2 r_n\right) \le 2E\exp(-2nr_n^2), \tag{66}$$

$$\Pr\left(\max_{i,j\in[E]}|\widehat{G}_{ij} - G_{ij}| > B_f^2 r_n\right) \le 2E^2\exp\left(-\frac{nr_n^2}{2}\right). \tag{67}$$

By the definition of r_n and a union bound over these four events, with probability at least $1-\delta$ we are on an event Ω such that

$$\max_{e\in[E]}|\widehat{q}_e - q_e| \le r_n, \qquad \max_{e\in[E]}|\widehat{a}_e - a_e| \le r_n, \tag{68}$$

$$\max_{e\in[E]}|\widehat{v}_e - v_e| \le B_f^2 r_n, \qquad \max_{i,j\in[E]}|\widehat{G}_{ij} - G_{ij}| \le B_f^2 r_n. \tag{69}$$

Step 2: Concentration of the empirical CP and ACP scores. Fix the event Ω. Since $r_n \le q_{\min}/2$ and $q_e \ge q_{\min}$, we have $\widehat{q}_e \ge q_{\min}/2 > 0$ for every e, so $\widehat{\mathrm{CP}}_e$ is well-defined on Ω. Moreover, $0 \le a_e \le q_e$ because $0 \le p^{(e)}(t) \le 1$, so

$$|\widehat{\mathrm{CP}}_e - \mathrm{CP}_e| = \left|\frac{\widehat{a}_e}{\widehat{q}_e} - \frac{a_e}{q_e}\right| \tag{70}$$

$$\le \frac{|\widehat{a}_e - a_e|}{\widehat{q}_e} + a_e\left|\frac{1}{\widehat{q}_e} - \frac{1}{q_e}\right| \tag{71}$$

$$\le \frac{2r_n}{q_{\min}} + q_e\frac{|\widehat{q}_e - q_e|}{q_e\widehat{q}_e} \tag{72}$$

$$\le \frac{2r_n}{q_{\min}} + \frac{2r_n}{q_{\min}} = C_{\mathrm{CP}} r_n. \tag{73}$$

This proves the score concentration bound for DO-CP.

In the DO-ACP case, the condition $B_f^2 r_n \le v_{\min}/2$ implies $\widehat{v}_e \ge v_{\min}/2$ for every e. Since the derivative of $x \mapsto \sqrt{x}$ is bounded by $1/\sqrt{2v_{\min}}$ on $[v_{\min}/2, \infty)$, the mean-value theorem gives

$$|\sqrt{\widehat{v}_e} - \sqrt{v_e}| \le \frac{|\widehat{v}_e - v_e|}{\sqrt{2v_{\min}}} \le \frac{B_f^2}{\sqrt{2v_{\min}}} r_n. \tag{74}$$

Also $\widehat{v}_e \le B_f^2$, hence $\sqrt{\widehat{v}_e} \le B_f$, and $\mathrm{CP}_e \le 1$. Therefore

$$|\widehat{\mathrm{ACP}}_e - \mathrm{ACP}_e| = \left|\widehat{\mathrm{CP}}_e\sqrt{\widehat{v}_e} - \mathrm{CP}_e\sqrt{v_e}\right| \tag{75}$$

$$\le \sqrt{\widehat{v}_e}\,|\widehat{\mathrm{CP}}_e - \mathrm{CP}_e| + \mathrm{CP}_e\,|\sqrt{\widehat{v}_e} - \sqrt{v_e}| \tag{76}$$

$$\le B_f C_{\mathrm{CP}} r_n + \frac{B_f^2}{\sqrt{2v_{\min}}} r_n = C_{\mathrm{ACP}} r_n. \tag{77}$$

Thus, in either score choice,

$$\max_{e\in[E]}|\widehat{I}_e - I_e| \le C_I r_n. \tag{78}$$

Step 3: Entrywise kernel concentration. Because $C_I r_n \le I_{\min}/2$ and $I_e \ge I_{\min}$, we have

$$\widehat{I}_e \in \left[\frac{I_{\min}}{2},\, I_{\max} + \frac{I_{\min}}{2}\right] \qquad \text{for every } e \in [E]. \tag{79}$$

Consider the map $h(a, b, c) := \sqrt{ab}\, c$ on the compact domain

$$D := \left[\frac{I_{\min}}{2},\, I_{\max} + \frac{I_{\min}}{2}\right]^2 \times [-B_f^2, B_f^2]. \tag{80}$$

On D, its partial derivatives satisfy

$$\left|\frac{\partial h}{\partial a}\right|, \left|\frac{\partial h}{\partial b}\right| \le \frac{B_f^2}{2}\sqrt{\frac{2I_{\max}}{I_{\min}} + 1}, \tag{81}$$

$$\left|\frac{\partial h}{\partial c}\right| \le I_{\max} + \frac{I_{\min}}{2}. \tag{82}$$

By the mean-value theorem,

$$|\widehat{\mathcal{K}}_{ij} - \mathcal{K}_{ij}| = |h(\widehat{I}_i, \widehat{I}_j, \widehat{G}_{ij}) - h(I_i, I_j, G_{ij})| \tag{83}$$

$$\le C_h \max\Big\{|\widehat{I}_i - I_i|, |\widehat{I}_j - I_j|, |\widehat{G}_{ij} - G_{ij}|\Big\} \tag{84}$$

$$\le C_h \max\{C_I, B_f^2\}\, r_n = \varepsilon_n \tag{85}$$

for all $i, j \in [E]$.

Step 4: Transfer to the log-det objective. The diagonal regularizer is stable under the same entrywise bound:

$$|\widehat{\lambda}_{\mathrm{reg}} - \lambda_{\mathrm{reg}}| \le \frac{\varepsilon_n}{K}. \tag{86}$$

For any S with $|S| = K$, the matrix $\widehat{\boldsymbol{\mathcal{K}}}_S - \boldsymbol{\mathcal{K}}_S$ is $K \times K$ and has entrywise bound ε_n, so

$$\left\|\widehat{\boldsymbol{\mathcal{K}}}_S + \widehat{\lambda}_{\mathrm{reg}}\mathbf{I} - \boldsymbol{\mathcal{K}}_S - \lambda_{\mathrm{reg}}\mathbf{I}\right\|_{\mathrm{op}} \le K\varepsilon_n + \frac{\varepsilon_n}{K} = \Delta_n. \tag{87}$$

Let $\sigma_1, \ldots, \sigma_K$ be the eigenvalues of $\boldsymbol{\mathcal{K}}_S + \lambda\mathbf{I}$ and $\widehat{\sigma}_1, \ldots, \widehat{\sigma}_K$ those of $\widehat{\boldsymbol{\mathcal{K}}}_S + \widehat{\lambda}_{\mathrm{reg}}\mathbf{I}$. Weyl's inequality gives

$$|\widehat{\sigma}_i - \sigma_i| \le \Delta_n \qquad \text{for each } i \in [K]. \tag{88}$$

Since $\Delta_n < \lambda$, every $\widehat{\sigma}_i$ is positive. Therefore

$$\widehat{F}(S) - F(S) = \sum_{i=1}^{K} \log\left(\frac{\widehat{\sigma}_i}{\sigma_i}\right) = \sum_{i=1}^{K} \log\left(1 + \frac{\widehat{\sigma}_i - \sigma_i}{\sigma_i}\right). \tag{89}$$

Because $\sigma_i \ge \lambda$ and $|\widehat{\sigma}_i - \sigma_i| \le \Delta_n$, we have

$$K \log\left(1 - \frac{\Delta_n}{\lambda}\right) \le \widehat{F}(S) - F(S) \le K \log\left(1 + \frac{\Delta_n}{\lambda}\right). \tag{90}$$

Taking absolute values yields

$$\left|\widehat{F}(S) - F(S)\right| \le -K \log\left(1 - \frac{\Delta_n}{\lambda}\right). \tag{91}$$

Applying the one-sided bounds to $\widehat{S}$ and $S^\star$ gives

$$F(\widehat{S}) \ge \widehat{F}(\widehat{S}) - K \log\left(1 + \frac{\Delta_n}{\lambda}\right) \tag{92}$$

$$\ge \widehat{F}(S^\star) - K \log\left(1 + \frac{\Delta_n}{\lambda}\right) \tag{93}$$

$$\ge F(S^\star) + K \log\left(1 - \frac{\Delta_n}{\lambda}\right) - K \log\left(1 + \frac{\Delta_n}{\lambda}\right), \tag{94}$$

which is exactly the claimed transfer bound. □

G.2 Grouping recovery

This subsection analyzes the specific OC variant defined in Appendix H: average-linkage agglomerative clustering on empirical cosine dissimilarities.

Fix a selected expert set $S \subseteq [E]$ with $|S| = K$ and a target group count G. Define the population second moments

$$M_{ij} := \mathbb{E}[\langle f_i(t), f_j(t) \rangle], \qquad i, j \in S, \tag{95}$$

and assume throughout this subsection that

$$M_{ee} > 0 \qquad \text{for every } e \in S, \tag{96}$$

so that the population cosine similarities below are well defined. Define

$$\rho_{ij} := \frac{M_{ij}}{\sqrt{M_{ii} M_{jj}}}, \qquad i, j \in S, \tag{97}$$

the population cosine similarities, and

$$d(i, j) := 1 - \rho_{ij}. \tag{98}$$

For two nonempty disjoint clusters $C, C' \subseteq S$, define the average-linkage distance

$$d_{\mathrm{avg}}(C, C') := \frac{1}{|C|\,|C'|} \sum_{i \in C} \sum_{j \in C'} d(i, j). \tag{99}$$

Definition G.2 (Average-linkage output clustering)**.** *Starting from the singleton partition of S, repeatedly merge the pair of clusters with minimum $d_{\mathrm{avg}}(C, C')$. Stop when exactly G clusters remain.*

Assumption G.3 (Output-space separation)**.** *There exists a partition $\mathcal{P}^\star = \{\mathcal{G}_1^\star, \dots, \mathcal{G}_G^\star\}$ of S and constants $\Delta_{\mathrm{in}}, \Delta_{\mathrm{out}}$ with $\Delta_{\mathrm{in}} < \Delta_{\mathrm{out}}$ such that*

$$\max_{g \in [G]} \max_{i, j \in \mathcal{G}_g^\star} d(i, j) \le \Delta_{\mathrm{in}} < \Delta_{\mathrm{out}} \le \min_{g \neq h} \min_{\substack{i \in \mathcal{G}_g^\star \\ j \in \mathcal{G}_h^\star}} d(i, j). \tag{100}$$

Theorem G.4 (Exact recovery of output clustering)**.** *Assume Assumption G.3. Then average-linkage output clustering returns $\mathcal{P}^\star$ (up to permutation of the group labels).*

Proof. Step 1: Cluster-level separation. Let $C \subseteq \mathcal{G}_g^\star$ and $C' \subseteq \mathcal{G}_g^\star$ be disjoint clusters contained in the same true group. Then every pair $(i, j) \in C \times C'$ satisfies $d(i, j) \le \Delta_{\mathrm{in}}$, hence

$$d_{\mathrm{avg}}(C, C') \le \Delta_{\mathrm{in}}. \tag{101}$$

Similarly, if $C \subseteq \mathcal{G}_g^\star$ and $C' \subseteq \mathcal{G}_h^\star$ with $g \neq h$, then every pair $(i, j) \in C \times C'$ satisfies $d(i, j) \ge \Delta_{\mathrm{out}}$, so

$$d_{\mathrm{avg}}(C, C') \ge \Delta_{\mathrm{out}}. \tag{102}$$

Step 2: Induction on the merge steps. Initially every cluster is a singleton and is therefore contained in some true group. Suppose inductively that, at a given iteration, every current cluster is contained in a true group. If a true group $\mathcal{G}_g^\star$ is represented by at least two current clusters $C_1, C_2 \subseteq \mathcal{G}_g^\star$, then Step 1 gives

$$d_{\mathrm{avg}}(C_1, C_2) \le \Delta_{\mathrm{in}}. \tag{103}$$

By contrast, any pair of clusters drawn from different true groups has average-linkage distance at least Δ_{out}. Since $\Delta_{\mathrm{in}} < \Delta_{\mathrm{out}}$, the globally closest pair must lie within the same true group. Therefore the algorithm performs an intra-group merge, and the merged cluster remains contained in a true group. This preserves the induction hypothesis.

Step 3: Termination. By Step 2, the algorithm performs only intra-group merges until each true group has been merged into a single cluster. At that moment exactly G clusters remain, namely $\mathcal{G}_1^\star, \dots, \mathcal{G}_G^\star$. □

Assumption G.5 (Bounded outputs and nondegenerate norms for output clustering)**.** *There exist constants $B_f, \sigma_{\min} > 0$ such that*

$$\|f_e(t)\|_2 \le B_f \quad \textit{almost surely} \qquad \textit{and} \qquad M_{ee} \ge \sigma_{\min}^2 \tag{104}$$

for every $e \in S$.

Lemma G.6 (Uniform concentration of empirical cosine similarities)**.** *Assume Assumption G.5. Given i.i.d. calibration tokens $t_1, \dots, t_n \sim \mathcal{D}$, define*

$$\widehat{M}_{ij} := \frac{1}{n} \sum_{m=1}^{n} \langle f_i(t_m), f_j(t_m) \rangle, \tag{105}$$

$$\widehat{\rho}_{ij} := \begin{cases} \widehat{M}_{ij} / \sqrt{\widehat{M}_{ii} \widehat{M}_{jj}}, & \widehat{M}_{ii} \widehat{M}_{jj} > 0, \\ 0, & \widehat{M}_{ii} \widehat{M}_{jj} = 0, \end{cases} \tag{106}$$

$$\widehat{d}(i,j) := 1 - \widehat{\rho}_{ij}. \tag{107}$$

Let

$$C_{\mathrm{oc}} := \frac{2}{\sigma_{\min}^2} + \frac{4B_f^2}{\sigma_{\min}^4}. \tag{108}$$

Then for any $\tau \in (0, \sigma_{\min}^2/2]$,

$$\Pr\left(\max_{i,j \in S} |\widehat{M}_{ij} - M_{ij}| > \tau \textit{ or } \max_{i,j \in S} |\widehat{\rho}_{ij} - \rho_{ij}| > C_{\mathrm{oc}} \tau \right) \le 2|S|^2 \exp\left(-\frac{n\tau^2}{2B_f^4} \right). \tag{109}$$

Proof. Fix $i, j \in S$ and define $X_m^{(ij)} := \langle f_i(t_m), f_j(t_m) \rangle$. By Cauchy–Schwarz and Assumption G.5,

$$|X_m^{(ij)}| \le \|f_i(t_m)\|_2 \|f_j(t_m)\|_2 \le B_f^2 \tag{110}$$

almost surely, and $\mathbb{E}[X_m^{(ij)}] = M_{ij}$. Hoeffding's inequality therefore gives

$$\Pr\left(|\widehat{M}_{ij} - M_{ij}| > \tau \right) \le 2 \exp\left(-\frac{n\tau^2}{2B_f^4} \right). \tag{111}$$

Applying a union bound over $(i,j) \in S \times S$, we obtain an event Ω_τ of probability at least

$$1 - 2|S|^2 \exp\left(-\frac{n\tau^2}{2B_f^4} \right) \tag{112}$$

on which

$$\max_{i,j \in S} |\widehat{M}_{ij} - M_{ij}| \le \tau. \tag{113}$$

Fix this event. Since $M_{ii} \ge \sigma_{\min}^2$ and $\tau \le \sigma_{\min}^2/2$, we have

$$\widehat{M}_{ii} \ge M_{ii} - \tau \ge \sigma_{\min}^2/2 > 0 \tag{114}$$

for every $i \in S$. Hence $\widehat{\rho}_{ij} = \widehat{M}_{ij} / \sqrt{\widehat{M}_{ii} \widehat{M}_{jj}}$ on Ω_τ. Now fix $i, j \in S$. Using $\sqrt{\widehat{M}_{ii} \widehat{M}_{jj}} \ge \sigma_{\min}^2/2$,

$$|\widehat{\rho}_{ij} - \rho_{ij}| \le \frac{|\widehat{M}_{ij} - M_{ij}|}{\sqrt{\widehat{M}_{ii} \widehat{M}_{jj}}} + |M_{ij}| \left| \frac{1}{\sqrt{\widehat{M}_{ii} \widehat{M}_{jj}}} - \frac{1}{\sqrt{M_{ii} M_{jj}}} \right| \tag{115}$$

$$\le \frac{2\tau}{\sigma_{\min}^2} + B_f^2 \left| \frac{1}{\sqrt{\widehat{M}_{ii} \widehat{M}_{jj}}} - \frac{1}{\sqrt{M_{ii} M_{jj}}} \right|. \tag{116}$$

For the reciprocal term,

$$\left|\frac{1}{\sqrt{\widehat{M}_{ii}\widehat{M}_{jj}}}-\frac{1}{\sqrt{M_{ii}M_{jj}}}\right|\le\frac{1}{\sqrt{\widehat{M}_{jj}}}\left|\frac{1}{\sqrt{\widehat{M}_{ii}}}-\frac{1}{\sqrt{M_{ii}}}\right|+\frac{1}{\sqrt{M_{ii}}}\left|\frac{1}{\sqrt{\widehat{M}_{jj}}}-\frac{1}{\sqrt{M_{jj}}}\right|. \quad (117)$$

Since $x\mapsto x^{-1/2}$ has derivative $-\frac{1}{2}x^{-3/2}$, it is $\sqrt{2}/\sigma_{\min}^3$-Lipschitz on $[\sigma_{\min}^2/2,\infty)$. Therefore

$$\left|\frac{1}{\sqrt{\widehat{M}_{ii}}}-\frac{1}{\sqrt{M_{ii}}}\right|,\left|\frac{1}{\sqrt{\widehat{M}_{jj}}}-\frac{1}{\sqrt{M_{jj}}}\right|\le\frac{\sqrt{2}}{\sigma_{\min}^3}\tau. \quad (118)$$

Using $1/\sqrt{\widehat{M}_{jj}}\le\sqrt{2}/\sigma_{\min}$ and $1/\sqrt{M_{ii}}\le 1/\sigma_{\min}$, we obtain

$$\left|\frac{1}{\sqrt{\widehat{M}_{ii}\widehat{M}_{jj}}}-\frac{1}{\sqrt{M_{ii}M_{jj}}}\right|\le\frac{2+\sqrt{2}}{\sigma_{\min}^4}\tau\le\frac{4}{\sigma_{\min}^4}\tau. \quad (119)$$

Combining the preceding bounds yields

$$|\widehat{\rho}_{ij}-\rho_{ij}|\le\left(\frac{2}{\sigma_{\min}^2}+\frac{4B_f^2}{\sigma_{\min}^4}\right)\tau=C_{\mathrm{oc}}\tau. \quad (120)$$

This holds uniformly over $i,j\in S$ on Ω_τ. □

Theorem G.7 (Finite-sample recovery of output clustering)**.** *Assume Assumptions G.3 and G.5. Let*

$$\gamma:=\Delta_{\mathrm{out}}-\Delta_{\mathrm{in}}>0,\qquad\tau_\star:=\min\left\{\frac{\sigma_{\min}^2}{2},\frac{\gamma}{4C_{\mathrm{oc}}}\right\}. \quad (121)$$

If

$$n\ge\frac{2B_f^4}{\tau_\star^2}\log\left(\frac{2|S|^2}{\delta}\right), \quad (122)$$

then average-linkage output clustering on the empirical dissimilarities $\widehat{d}(i,j)$ returns $\mathcal{P}^\star$ with probability at least $1-\delta$.

Proof. By Lemma G.6 with $\tau=\tau_\star$, the stated lower bound on n guarantees that with probability at least $1-\delta$,

$$\max_{i,j\in S}|\widehat{\rho}_{ij}-\rho_{ij}|\le C_{\mathrm{oc}}\tau_\star\le\frac{\gamma}{4}. \quad (123)$$

Fix this event and set $\varepsilon:=\gamma/4$. Then

$$|\widehat{d}(i,j)-d(i,j)|=|\widehat{\rho}_{ij}-\rho_{ij}|\le\varepsilon\qquad\text{for all } i,j\in S. \quad (124)$$

Hence, if i,j lie in the same true group, Assumption G.3 gives

$$\widehat{d}(i,j)\le d(i,j)+\varepsilon\le\Delta_{\mathrm{in}}+\varepsilon. \quad (125)$$

If i,j lie in different true groups, then

$$\widehat{d}(i,j)\ge d(i,j)-\varepsilon\ge\Delta_{\mathrm{out}}-\varepsilon. \quad (126)$$

Since $\varepsilon=\gamma/4$, we have

$$\Delta_{\mathrm{in}}+\varepsilon<\Delta_{\mathrm{out}}-\varepsilon. \quad (127)$$

Thus the empirical dissimilarity $\widehat{d}$ satisfies the same strict separation property as in Assumption G.3, with thresholds $\Delta_{\mathrm{in}}+\varepsilon$ and $\Delta_{\mathrm{out}}-\varepsilon$. Theorem G.4 therefore applies to $\widehat{d}$ and yields exact recovery of $\mathcal{P}^\star$. □

G.3 Merging optimality

Fix a selected subset $S \subseteq [E]$ and a partition $\{\mathcal{G}_g\}_{g=1}^G$ of S. View each expert output f_e as an element of the Hilbert space

$$\mathcal{H} := L^2(\mathcal{D}; \mathbb{R}^d) \tag{128}$$

with inner product

$$\langle u, v \rangle_{\mathcal{H}} := \mathbb{E}_{t \sim \mathcal{D}}[\langle u(t), v(t) \rangle]. \tag{129}$$

Theorem G.8 (Score-weighted output averaging is the oracle proxy representative)**.** *Let $s^{(e)} \geq 0$ be the expert scores used for within-group weighting, and assume that every group has strictly positive score mass*

$$S_g := \sum_{e \in \mathcal{G}_g} s^{(e)} > 0. \tag{130}$$

For group representatives $\mu_1, \ldots, \mu_G \in \mathcal{H}$, define the quadratic merge distortion

$$\mathcal{L}(\mu_1, \ldots, \mu_G) := \sum_{g=1}^{G} \sum_{e \in \mathcal{G}_g} s^{(e)} \|f_e - \mu_g\|_{\mathcal{H}}^2. \tag{131}$$

Then the unique minimizer is

$$\mu_g^\star := \frac{1}{S_g} \sum_{e \in \mathcal{G}_g} s^{(e)} f_e, \qquad g \in [G]. \tag{132}$$

Thus, if merging were performed directly in output-function space, the oracle representative would use the same score weights that Eq. (1) *uses in parameter space.*

Proof. Fix a group g and write

$$L_g(\mu) := \sum_{e \in \mathcal{G}_g} s^{(e)} \|f_e - \mu\|_{\mathcal{H}}^2. \tag{133}$$

Set $\mu_g^\star := S_g^{-1} \sum_{e \in \mathcal{G}_g} s^{(e)} f_e$. For any $\mu \in \mathcal{H}$, expand

$$f_e - \mu = (f_e - \mu_g^\star) + (\mu_g^\star - \mu), \tag{134}$$

so

$$\|f_e - \mu\|_{\mathcal{H}}^2 = \|f_e - \mu_g^\star\|_{\mathcal{H}}^2 + \|\mu_g^\star - \mu\|_{\mathcal{H}}^2 + 2\langle f_e - \mu_g^\star, \mu_g^\star - \mu \rangle_{\mathcal{H}}. \tag{135}$$

Multiply by $s^{(e)}$ and sum over $e \in \mathcal{G}_g$:

$$\begin{aligned} L_g(\mu) = \sum_{e \in \mathcal{G}_g} s^{(e)} \|f_e - \mu_g^\star\|_{\mathcal{H}}^2 + S_g \|\mu_g^\star - \mu\|_{\mathcal{H}}^2 \\ + 2 \left\langle \sum_{e \in \mathcal{G}_g} s^{(e)} (f_e - \mu_g^\star), \mu_g^\star - \mu \right\rangle_{\mathcal{H}}. \end{aligned} \tag{136}$$

The last inner-product term vanishes by the definition of $\mu_g^\star$, giving

$$L_g(\mu) = L_g(\mu_g^\star) + S_g \|\mu - \mu_g^\star\|_{\mathcal{H}}^2. \tag{137}$$

Since $S_g > 0$, the unique minimizer of L_g is $\mu_g^\star$. Summing this identity over $g = 1, \ldots, G$ proves that $(\mu_g^\star)_{g=1}^G$ uniquely minimizes $\mathcal{L}$. □

Remark G.9 (Parameter-space averaging is a proxy)**.** *Theorem G.8 is a statement about output functions in $\mathcal{H}$. It does not imply that averaging the parameters of nonlinear SwiGLU experts exactly realizes $\mu_g^\star$. The implemented merge in Eq.* (1) *applies the oracle function-space weights to parameters as a practical proxy; it is exact only in special cases such as linear experts or singleton groups.*

Proposition G.10 (Scaling induces explicit global weights in the function-space proxy)**.** *Let*

$$w^{(e)} := \frac{s^{(e)}}{S_g}, \qquad e \in \mathcal{G}_g, \tag{138}$$

be the score weights from Eq. (1)*, define the function-space proxy representative*

$$\mu_g^{\text{proxy}} := \sum_{e \in \mathcal{G}_g} w^{(e)} f_e, \tag{139}$$

and let $\alpha_1, \ldots, \alpha_G \geq 0$ be arbitrary group scaling factors. Then the scaled proxy output can be written exactly as

$$\sum_{g=1}^{G} \alpha_g \mu_g^{\text{proxy}} = \sum_{e \in S} \eta_e f_e, \qquad \eta_e := \alpha_{g(e)} \frac{s^{(e)}}{S_{g(e)}}, \tag{140}$$

where $g(e)$ is the unique group index such that $e \in \mathcal{G}_{g(e)}$.

Proof. By definition of $w^{(e)}$,

$$\sum_{g=1}^{G} \alpha_g \mu_g^{\text{proxy}} = \sum_{g=1}^{G} \alpha_g \left(\sum_{e \in \mathcal{G}_g} \frac{s^{(e)}}{S_g} f_e \right) \tag{141}$$

$$= \sum_{g=1}^{G} \sum_{e \in \mathcal{G}_g} \alpha_g \frac{s^{(e)}}{S_g} f_e \tag{142}$$

$$= \sum_{e \in S} \alpha_{g(e)} \frac{s^{(e)}}{S_{g(e)}} f_e, \tag{143}$$

because $\{\mathcal{G}_g\}_{g=1}^{G}$ partitions S. □

Corollary G.11 (Proportional scaling recovers global score weighting in the proxy)**.** *Under proportional scaling,*

$$\alpha_g = \frac{S_g}{\sum_{e' \in S} s^{(e')}}, \tag{144}$$

the induced global weights are

$$\eta_e = \frac{s^{(e)}}{\sum_{e' \in S} s^{(e')}}. \tag{145}$$

Thus proportional scaling makes the scaled proxy output exactly equal to the global score-weighted average over the selected experts.

Proof. Substitute the proportional-scaling formula into Proposition G.10:

$$\eta_e = \frac{S_{g(e)}}{\sum_{e' \in S} s^{(e')}} \cdot \frac{s^{(e)}}{S_{g(e)}} = \frac{s^{(e)}}{\sum_{e' \in S} s^{(e')}}. \tag{146}$$

□

Corollary G.12 (Uniform scaling equalizes group mass in the proxy)**.** *Under uniform scaling,*

$$\alpha_g = \frac{1}{G}, \tag{147}$$

the induced global weights are

$$\eta_e = \frac{s^{(e)}}{G\, S_{g(e)}}, \tag{148}$$

each group's total weight is exactly

$$\sum_{e \in \mathcal{G}_g} \eta_e = \frac{1}{G}, \tag{149}$$

and the total weight across all selected experts sums to 1*, matching the simplex constraint of the original MoE router.*

Proof. The first identity is Proposition G.10 with $\alpha_g = 1/G$. Summing over one group gives

$$\sum_{e \in \mathcal{G}_g} \eta_e = \frac{1}{G\, S_g} \sum_{e \in \mathcal{G}_g} s^{(e)} = \frac{1}{G}, \tag{150}$$

and summing over all G groups yields $\sum_{e \in S} \eta_e = 1$. □

H Grouping strategy definitions

Given K selected experts assigned to k groups:

Round-robin (RR). Experts are sorted by descending score and assigned cyclically: expert with rank r goes to group $r \bmod G$. This produces balanced groups by construction.

Weight clustering (WC). Agglomerative clustering on the concatenated flattened gate, up, and down projection matrices ($\sim$4.7M dimensions) with cosine similarity.

Router clustering (RC). Agglomerative clustering on router gate weight vectors [24], using cosine similarity in the d-dimensional router space.

Anchor-based (AB). The k highest-scoring experts serve as anchors; remaining experts are assigned to the anchor with highest router-vector cosine similarity [24].

Output clustering (OC). Average-linkage agglomerative clustering on the empirical cosine dissimilarities. Let $\bar{G}_{ij} = \frac{1}{n}\sum_{m=1}^{n} \langle f_i(t_m), f_j(t_m) \rangle$ and $\bar{V}_i = \frac{1}{n}\sum_{m=1}^{n} \|f_i(t_m)\|_2^2$. Then:

$$\widehat{\rho}_{ij} := \begin{cases} \bar{G}_{ij} / \sqrt{\bar{V}_i\, \bar{V}_j}, & \bar{V}_i\, \bar{V}_j > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{151}$$

$$\widehat{d}(i,j) := 1 - \widehat{\rho}_{ij}, \tag{152}$$

computed from the calibration tokens $t_1, \ldots, t_n$. HC-SMoE [5] showed output-based clustering substantially outperforms weight and router similarity for grouping.

I Down-projection scaling equations

After merging, each group's down-projection is scaled by α_g.

Uniform scaling. Each group is scaled equally so that the total contribution sums to 1, matching the simplex constraint of the MoE router:

$$\tilde{\mathbf{W}}^{(g)}_{\text{down}} = \frac{1}{k}\, \mathbf{W}^{(g)}_{\text{down}}. \tag{153}$$

Proportional scaling.

$$\tilde{\mathbf{W}}^{(g)}_{\text{down}} = \frac{\sum_{e \in \mathcal{G}_g} s^{(e)}_{\ell}}{\sum_{e' \in S} s^{(e')}_{\ell}}\, \mathbf{W}^{(g)}_{\text{down}}, \tag{154}$$

where S is the set of K selected experts. Proportional scaling preserves relative contribution magnitude.

J Knowledge distillation: setup and method exploration

Forward KL training loss. The concatenated dense model provides an initialization for knowledge distillation from the MoE teacher. We minimize the forward KL divergence between the teacher and student output distributions:

$$\mathcal{L}_{\text{KD}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} D_{\text{KL}}\big(p_{\text{teacher}}(\cdot \mid x_t) \,\|\, p_{\text{student}}(\cdot \mid x_t) \big), \tag{155}$$

where $\mathcal{T}$ is the set of tokens in a batch and x_t is the context preceding token t. Training hyperparameters are given in Section 3.1.

Starting from the best initialization (DO-ACP, K=8), we ablate two distillation choices on top of this default loss.

Expanded teacher routing. Kim et al. [20] show that non-activated MoE experts contain useful knowledge for distillation. We test overriding the teacher's routing to activate $k' > k$ experts during distillation:

$$F_{\text{teacher}}(t) = \sum_{e \in \mathcal{S}_{k'}(t)} p_\ell^{(e)}(t)\, f_e(t), \tag{156}$$

where $\mathcal{S}_{k'}(t)$ is the set of k' highest-probability experts for token t. This exposes the student to knowledge from experts the router would not normally select, at the cost of $\sim k'/k$ times more expert activations per MoE layer. We sweep $k' \in \{8, 16, 32, 64, 96, 128\}$. Consistent with Kim et al. [20], an intermediate routing breadth (k'=16, i.e., $2k$) improves quality by +0.70 pp (Table 6), likely because the next-highest-scoring experts provide complementary knowledge. Beyond k'=32, performance degrades monotonically, likely because low-ranked experts add noise. However, the gain is modest relative to the $\sim 2\times$ teacher FLOPs per MoE layer, so we keep standard routing by default.

Table 6: Effect of expanded teacher routing during distillation (0.3B tokens each). All configurations use the same pruned student (DO-ACP). An intermediate k'=16 yields the best accuracy (+0.70 pp), not the default 8 or all 128 experts.

k'	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)	Δ
8 (default)	57.0	**41.1**	57.4	29.9	31.7	43.41	–
16	**57.9**	40.9	**57.8**	**31.1**	**32.8**	**44.11**	+0.70
32	57.1	40.6	57.0	31.0	32.5	43.63	+0.22
64	55.0	39.5	56.1	30.8	32.1	42.71	−0.70
96	56.0	38.8	55.8	29.8	31.6	42.39	−1.02
128 (all)	55.6	38.3	54.5	29.0	31.6	41.78	−1.63

Loss function ablation. We compare forward KL (baseline), reverse KL, and forward KL augmented with intermediate hidden-state MSE loss [31]. Forward KL substantially outperforms both alternatives (Table 7). Reverse KL loses −6.24 pp, likely because the student benefits more by recovering the full teacher distribution rather than concentrating on high-probability modes. Adding intermediate hidden-state MSE also hurts (−1.91 pp). This is consistent with Minitron's recommendation of logit-only distillation when depth is not significantly reduced, as is the case for our student [31]. For the extended training in Section 3.3, we use forward KL with standard teacher routing (k'=k) to maximize training throughput.

Table 7: Loss function ablation (0.3B tokens each). All use the same pruned student (DO-ACP). Forward KL is the clear winner, while reverse KL and intermediate loss both degrade quality.

Loss	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)	Δ
Forward KL	**57.0**	**41.1**	**57.4**	**29.9**	**31.7**	**43.41**	–
Intermediate (logit + hidden MSE)	55.5	39.1	53.3	28.7	30.9	41.50	−1.91
Reverse KL	53.0	33.8	42.4	25.6	31.0	37.17	−6.24

K Qualitative analysis

To complement the benchmark evaluation, we analyze the generation quality of the four model checkpoints from Section 3.3. Each model is prompted on 567 MMLU samples (10 per subject, 57 subjects) with a chain-of-thought zero-shot format and generates free-form responses (temperature 0.7, up to 2048 tokens). We classify each response into six categories using rule-based heuristics for surface-level failures and LLM-as-a-judge (Claude Opus 4.6) for semantic errors: *incoherent* (nonsensical output), *repetitive loop* (same phrases cycling without progress), *knowledge error* (coherent but factually wrong), *reasoning error* (flawed logic in STEM), and *other* (topic drift, truncation, out-of-range).

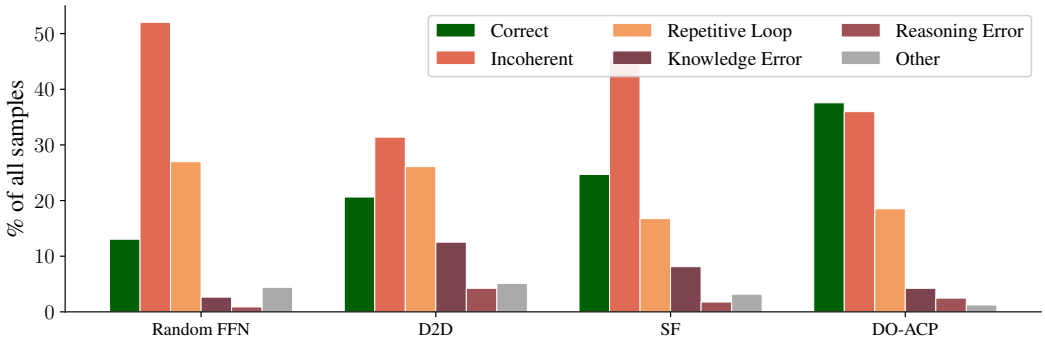


Figure 7: Error distribution on MMLU chain-of-thought zero-shot (567 samples, 4B-token models). As model quality improves (left to right), catastrophic failures (incoherent, repetitive) decrease and the correct rate rises. DO-ACP achieves the lowest total failure rate while maintaining the lowest knowledge error rate.

Catastrophic failures. Figure 7 shows the error distribution. We define catastrophic failures as responses that never reach meaningful reasoning: incoherent outputs and repetitive loops. DO-ACP has the lowest total catastrophic failure rate (54.5%), followed by D2D (57.5%), SF (62.3%), and Random FFN (79.0%). The pattern differs by failure mode: D2D has the lowest incoherent rate (31.4%), while SF has the lowest repetitive loop rate (16.8%). DO-ACP achieves the best overall rate by reducing both failure modes simultaneously.

Knowledge errors. Random FFN produces almost no knowledge errors (2.6%) because the vast majority of its outputs are catastrophic failures that never reach the stage of factual reasoning. Among the three models that do attempt reasoning, knowledge error rates differ substantially: D2D shows 12.5%, SF shows 8.1%, and DO-ACP shows only 4.2%. DO-ACP achieves the highest accuracy (37.6%) with the lowest knowledge error rate, indicating that diversity-aware expert selection preserves factual knowledge more effectively than both dense-to-dense pruning and frequency-based expert selection.

Subject-level breakdown. Breaking down accuracy by MMLU subject category confirms this pattern. DO-ACP reaches 49.2% on humanities, compared to 25.4% (SF), 19.2% (D2D), and 18.5% (Random FFN). The gap is largest on knowledge-intensive subjects (humanities +24 pp over SF, social sciences +11 pp) and smallest on STEM (+6 pp), where all models struggle with mathematical reasoning. This suggests that the experts selected by DO-ACP carry richer factual knowledge, consistent with the low knowledge error rate observed above.

Error taxonomy and examples. Table 8 defines the six error categories used above. Surface-level failures (incoherent, repetitive loop, other) are classified by rule-based heuristics applied in priority order; semantic errors (knowledge error, reasoning error) are classified by LLM-as-a-judge using Claude Opus 4.6. Table 9 shows representative examples from each category.

Table 8: Error taxonomy for qualitative MMLU analysis. Categories are applied in priority order (top to bottom): a response matching an earlier category is not checked against later ones.

Category	Definition
Incoherent	Nonsensical output that fails to convey meaning: circular redefinitions, meta-commentary without substance, or degenerate numeric sequences. Detected by filler-word ratio $>35\%$ and unique-word ratio $<15\%$, or digit ratio $>30\%$.
Repetitive Loop	The same phrases or sentences cycle repeatedly without making progress toward an answer. Detected by 3-gram repetition count ≥ 5.
Knowledge Error	The reasoning structure is coherent and on-topic, but the response contains factual errors (e.g., hallucinated definitions, incorrect attributions).
Reasoning Error	The response attempts logical or mathematical reasoning but arrives at an incorrect conclusion through flawed logic.
Other	Topic drift (coherent but off-topic, keyword overlap $<15\%$), truncation without a final answer, or out-of-range answer selection.

Table 9: Representative error examples from the qualitative MMLU analysis.

Category	Model	Example (excerpt)
Incoherent	Random FFN	"The duty is a principle, but the principle is a norm. The duty is a moral obligation, but the norm is a standard of conduct…" (circular redefinition, no content)
Incoherent	SF	"We need to find the correct answer. First, the first step is to check the list of options…the next question is: what is the correct fact?" (meta-commentary about answering without engaging the question)
Repetitive Loop	D2D	"Jaden's score is a multiple of 7. The number of points Jaden scored is less than 45. So, Jaden's score is a multiple of 7…" (same two premises repeated with no computation)
Knowledge Error	D2D	"Miaphystism is the doctrine that God is present in the Eucharist…mīaphysos is a word that can be translated as disease…" (hallucinated definition and fabricated etymology)
Reasoning Error	DO-ACP	"1/2 exponent is the smallest…1/2 is the fastest." Self-contradiction within two sentences; growth rate ordering reversed.
Other	Random FFN	Four answer choices paraphrased at equal length; response truncated mid-sentence without selecting A–D.

L Base vs. post-trained teacher comparison

All primary experiments use the post-trained (instruct/reasoning hybrid) variant of Qwen3-30B-A3B as the teacher model. To verify that this choice does not bias our findings, we repeat the full 350-configuration pre-distill PPL sweep and distill the top-1 configuration per scoring method at both $K{=}8$ and $K{=}16$ using the base variant (`Qwen3-30B-A3B-Base`) with identically collected importance scores and Gram matrices.

Scoring rankings are preserved across model variants. Table 10 compares the best pre-distill PPL per scoring method between the base and post-trained teacher variants. The ranking by best PPL is nearly identical:

The top-4 scoring methods (ACP, DO-CP, DO-ACP, CP) maintain their exact ranking across both variants. The bottom three (PP, PS, SF) swap positions 5–7 but remain tightly clustered. Base PPL values are systematically higher than post-trained values (1.2–1.6×).

Downstream benchmarks confirm PPL rankings. We distill the top-1 base configuration per scoring method at both $K{=}8$ (pure pruning) and $K{=}16$ (merging) for 0.3B tokens and evaluate on all five benchmarks. Table 11 shows results.

The three-tier scoring hierarchy from the main experiments is preserved: ACP and DO-ACP lead at $\sim$42.7%, followed by CP at $\sim$41.1%, then SF at 34–37%, and random baselines at 30–32%. Pure pruning ($K{=}8$) outperforms merging ($K{=}16$) for ACP (+2.68 pp) and DO-ACP (+1.66 pp),

Table 10: Best pre-distill PPL per scoring method: post-trained (instruct) vs. base teacher. Rankings match in the top-4 positions (ACP, DO-CP, DO-ACP, CP), confirming that scoring method conclusions are robust to teacher variant choice.

Scoring	Post-trained (Instruct)		Base	
	Best PPL	Rank	Best PPL	Rank
ACP	2,002	1	2,424	1
DO-CP	2,840	2	4,608	2
DO-ACP	3,939	3	5,697	3
CP	4,280	4	8,175	4
PP	9,463	5	9,326	5
PS	8,509	6	11,308	6
SF	8,638	7	13,370	7

Table 11: Base teacher distillation results (K=8 and K=16, 0.3B tokens each). **Bold**: best per scoring. ACP and DO-ACP at K=8 outperform their K=16 counterparts, confirming the "pure pruning wins" finding from the main experiments.

Scoring	K	Pre-distill PPL	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
ACP	**8**	9,049	54.8	**42.2**	**57.5**	**31.4**	28.1	**42.81**
ACP	16	2,424	54.5	39.2	53.5	28.0	25.4	40.13
DO-ACP	**8**	10,220	55.6	41.2	55.5	29.6	**31.0**	**42.59**
DO-ACP	16	5,697	54.6	39.6	53.7	28.6	28.1	40.93
CP	8	9,223	**55.6**	39.8	54.6	28.5	26.9	41.08
CP	16	8,175	54.9	37.7	56.9	29.5	26.3	41.07
SF	8	29,059	51.7	29.5	39.7	22.1	27.0	34.01
SF	16	13,370	52.6	33.4	48.4	24.1	27.2	37.17
Random FFN		–	51.9	28.0	33.0	22.3	26.1	32.26
Random init		–	50.4	25.2	28.0	23.8	22.9	30.08

confirming the finding from Section 3.2. CP shows a near-tie across K values, and only SF benefits from merging. Absolute accuracy is comparable between teacher variants: the best base configuration (ACP at K=8, 42.81%) slightly exceeds its instruct counterpart (42.52%, +0.29 pp), while DO-ACP at K=8 on the base model (42.59%) is 0.82 pp below instruct (43.41%).

Overall, scoring method rankings from the main experiments (Section 3.2) are robust to teacher variant, and the pure pruning advantage (Section 3.2) holds for both variants. This is consistent with Liu et al. [26], who found that post-trained teachers produce comparable or stronger students for knowledge distillation.

M Full distillation hyperparameters

Table 12 lists all hyperparameters used for knowledge distillation. Each configuration processes approximately 0.3B tokens ($200 \times 384 \times 4096$) and takes $\sim$5.5 hours on 4$\times$ H200 GPUs.

Table 12: Distillation hyperparameters for 200-step (0.3B-token) runs.

Parameter	**Value**
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$)
Weight decay	0.01
Peak learning rate	10^{-4}
Min learning rate	10^{-5}
LR schedule	Cosine decay (warmup 20 steps)
Temperature (τ)	1.0
Loss function	Forward KL divergence on logits
Sequence length	4096
Micro-batch size	4 per GPU
Gradient accumulation	24 steps
Global batch size	384 sequences
Total steps	200 (0.3B tokens)
Infrastructure	
GPUs	4$\times$ NVIDIA H200 (140 GB HBM3e each)
Parallelism	DeepSpeed ZeRO Stage 2
Precision	BF16 mixed precision
Attention	Flash Attention 2
Training data	FineWeb-Edu (streaming, no repeat)
Calibration data	512 samples from WikiText-103
Evaluation data	WikiText-2 test set (full)

N Full pre-distill PPL grid

Tables 13 and 14 present the comprehensive WikiText-2 perplexity for all 350 pre-distill configurations: 7 scoring $\times$ 5 grouping $\times$ 2 scaling (Uniform/Proportional) $\times$ 5 values of K. Both down-projection scaling options are shown separately for each scoring$\times$grouping pair. At K=8 ($= k$), each group contains exactly one expert; for DO-CP, ACP, and DO-ACP the same 8 experts are selected regardless of grouping, so all five grouping rows share the same value.

Table 13: WikiText-2 pre-distill perplexity (↓) for frequency-based and conditional probability scoring methods. Each row shows one scoring $\times$ grouping $\times$ DP scaling combination across all K values. **Bold**: best K per row. "–": numerical instability. Values $\geq$100k in compact notation.

Scoring	Grouping	DP Scaling	K=8	K=16	K=32	K=64	K=128
Selection Frequency	Round-Robin	Uniform	**20,633**	21,112	22,343	22,853	22,799
		Proportional	21,074	**19,440**	21,036	21,950	22,022
	Weight Cluster	Uniform	20,633	**12,896**	22,333	31,088	–
		Proportional	21,074	**17,661**	20,733	64,433	–
	Router Cluster	Uniform	20,633	**11,342**	14,833	13,473	14,607
		Proportional	21,074	**11,594**	18,269	30,705	80,436
	Anchor-Based	Uniform	**11,988**	15,494	16,796	17,916	17,963
		Proportional	**11,668**	13,863	15,362	16,385	16,661
	Output Cluster	Uniform	11,988	12,499	17,518	**8,638**	26,775
		Proportional	**11,668**	17,565	56,400	34,566	120k
Pre-Selection Probability	Round-Robin	Uniform	**14,983**	15,523	36,409	162k	315k
		Proportional	**13,340**	13,555	23,351	79,675	152k
	Weight Cluster	Uniform	14,983	**11,085**	12,528	19,148	20,430
		Proportional	13,340	**11,794**	22,710	74,225	430k
	Router Cluster	Uniform	14,983	14,665	**14,585**	15,038	15,176
		Proportional	13,340	**13,299**	30,706	82,825	122k
	Anchor-Based	Uniform	**9,463**	13,981	22,389	26,906	35,660
		Proportional	**10,697**	16,254	33,221	46,464	62,482
	Output Cluster	Uniform	**9,463**	10,223	36,539	61,045	44,465
		Proportional	**10,697**	18,176	77,920	206k	395k
Post-Selection Probability	Round-Robin	Uniform	18,027	**15,236**	15,276	15,405	15,378
		Proportional	**15,409**	15,996	16,281	16,178	15,976
	Weight Cluster	Uniform	18,027	**10,976**	13,436	15,552	–
		Proportional	15,409	**14,076**	16,900	20,448	–
	Router Cluster	Uniform	18,027	**11,379**	11,960	12,506	14,086
		Proportional	15,409	**8,850**	13,105	16,681	24,100
	Anchor-Based	Uniform	11,385	10,453	**10,425**	11,264	11,185
		Proportional	**12,269**	12,946	13,889	14,828	14,871
	Output Cluster	Uniform	11,385	10,914	16,675	**8,509**	25,862
		Proportional	**12,269**	15,557	41,130	24,286	62,593
Conditional Probability	Round-Robin	Uniform	**4,280**	20,580	311k	6.3m	21.8m
		Proportional	25,495	**6,846**	77,402	3.9m	18.5m
	Weight Cluster	Uniform	**4,280**	10,836	37,955	18,543	–
		Proportional	25,495	**11,730**	35,458	250k	–
	Router Cluster	Uniform	**4,280**	9,473	21,755	24,590	18,008
		Proportional	25,495	**5,068**	18,650	317k	1.6m
	Anchor-Based	Uniform	**6,033**	132k	54,246	92,860	170k
		Proportional	31,189	**25,609**	32,057	147k	348k
	Output Cluster	Uniform	**6,033**	6,085	12,998	55,860	186k
		Proportional	31,189	**6,897**	47,979	660k	956k

Table 14: WikiText-2 pre-distill perplexity (↓) for activation-weighted and D-optimal scoring methods. Each row shows one scoring × grouping × DP scaling combination across all K values. **Bold**: best K per row. Values ≥100k in compact notation.

Scoring	**Grouping**	**DP Scaling**	K=8	K=16	K=32	K=64	K=128
Activation-Wtd Cond. Prob.	Round-Robin	Uniform	**6,334**	13,080	67,926	417k	1.1m
		Proportional	31,273	31,129	**21,519**	32,031	129k
	Weight Cluster	Uniform	6,334	**5,527**	11,223	30,095	16,821
		Proportional	31,273	**16,018**	23,598	38,933	35,822
	Router Cluster	Uniform	6,334	**3,384**	7,586	9,161	15,296
		Proportional	31,273	10,450	**9,248**	19,192	203k
	Anchor-Based	Uniform	6,334	**4,829**	12,612	17,082	60,169
		Proportional	31,273	**7,273**	10,824	20,683	65,647
	Output Cluster	Uniform	6,334	**2,002**	12,079	26,600	151k
		Proportional	31,273	**30,198**	33,533	68,216	320k
D-Optimal + CP	Round-Robin	Uniform	**3,837**	5,137	60,925	1.1m	3.3m
		Proportional	10,928	**4,897**	65,655	1.1m	3.3m
	Weight Cluster	Uniform	**3,837**	4,970	17,150	36,424	16,186
		Proportional	10,928	**4,013**	41,808	260k	463k
	Router Cluster	Uniform	**3,837**	4,964	12,789	11,815	15,709
		Proportional	10,928	**6,050**	35,575	147k	728k
	Anchor-Based	Uniform	3,837	**2,840**	5,420	12,823	52,050
		Proportional	10,928	**3,022**	12,467	69,341	736k
	Output Cluster	Uniform	**3,837**	4,063	107k	42,441	113k
		Proportional	10,928	**3,805**	370k	333k	920k
D-Optimal + ACP	Round-Robin	Uniform	**5,134**	10,572	66,588	1.0m	2.8m
		Proportional	11,949	**10,945**	72,395	992k	2.8m
	Weight Cluster	Uniform	5,134	**4,779**	8,632	28,519	16,340
		Proportional	11,949	**4,333**	24,579	208k	431k
	Router Cluster	Uniform	**5,134**	8,553	8,751	9,410	15,635
		Proportional	11,949	**11,352**	34,707	109k	714k
	Anchor-Based	Uniform	5,134	**4,776**	15,564	18,842	50,751
		Proportional	11,949	**3,939**	16,248	78,899	743k
	Output Cluster	Uniform	**5,134**	5,521	22,932	102k	97,865
		Proportional	11,949	**6,230**	313k	552k	912k

O Full distillation results

Table 15 presents the full per-benchmark distillation results for all 35 scoring×grouping combinations on Qwen3-30B-A3B.

Table 15: Full distillation results (0.3B tokens) for all 35 configurations on Qwen3-30B-A3B. *Pre*/*Post* = WikiText-2 PPL before/after distillation. Color scale: best → mid → worst. **Bold**: best average per scoring method.

				PPL (↓)		Downstream Accuracy (%)					
Scoring	**Grouping**	**DP Scaling**	K	**Pre**	**Post**	**Wino**	**Hella**	**ARC-E**	**ARC-C**	**MMLU**	**Avg**
Selection Frequency	Round-robin	Proportional	16	19,440	29.3	53.04	32.58	49.07	25.51	27.62	37.56
	Weight cluster	Uniform	16	12,896	27.9	54.38	32.65	49.49	24.40	27.16	**37.62**
	Router cluster	Uniform	16	11,342	28.5	52.09	32.99	49.62	24.49	27.08	37.25
	Anchor-based	Proportional	8	11,668	29.2	52.17	31.64	47.14	23.12	27.50	36.31
	Output cluster	Uniform	64	8,638	27.7	53.12	33.06	49.33	24.74	27.34	37.52
Pre-Selection Probability	Round-robin	Proportional	8	13,340	30.9	53.12	31.85	48.53	23.89	27.57	36.99
	Weight cluster	Uniform	16	11,085	30.1	53.43	32.21	48.74	23.63	27.59	**37.12**
	Router cluster	Proportional	16	13,299	29.9	52.41	32.19	47.77	23.98	27.22	36.71
	Anchor-based	Uniform	8	9,463	28.6	53.35	32.17	48.65	24.15	27.28	**37.12**
	Output cluster	Uniform	8	9,463	28.7	52.88	32.14	47.47	23.63	27.70	36.77
Post-Selection Probability	Round-robin	Uniform	16	15,236	29.2	51.78	32.51	48.61	23.98	27.28	36.83
	Weight cluster	Uniform	16	10,976	27.7	52.64	32.84	49.79	24.66	27.07	**37.40**
	Router cluster	Proportional	16	8,850	28.8	51.85	32.68	49.07	25.09	27.33	37.21
	Anchor-based	Uniform	32	10,425	28.6	51.62	31.95	47.60	23.89	27.48	36.51
	Output cluster	Uniform	64	8,509	28.2	52.49	32.45	50.04	24.40	27.30	37.34
Conditional Probability	Round-robin	Proportional	16	3,696	24.4	53.83	36.18	56.36	27.82	27.60	40.35
	Weight cluster	Proportional	16	2,148	24.3	54.46	36.02	56.44	27.73	27.50	**40.43**
	Router cluster	Proportional	16	2,264	23.3	53.67	36.06	56.69	27.39	27.77	40.32
	Anchor-based	Uniform	8	6,033	21.5	53.67	35.73	55.51	26.96	27.11	39.80
	Output cluster	Uniform	8	6,033	21.5	53.83	35.73	55.72	27.82	26.80	39.98
Activation-Wtd Cond. Prob.	Round-robin	Uniform	8	6,334	19.1	56.35	39.78	56.02	29.69	30.73	**42.52**
	Weight cluster	Uniform	16	5,527	21.3	55.01	38.67	53.83	29.61	28.91	41.21
	Router cluster	Uniform	16	3,384	21.3	54.54	36.95	53.11	25.94	28.24	39.76
	Anchor-based	Uniform	16	4,829	21.7	52.01	37.30	53.41	26.88	27.64	39.45
	Output cluster	Uniform	16	2,002	20.5	54.78	38.11	55.05	27.47	27.11	40.50
D-Optimal + CP	Round-robin	Uniform	8	3,837	19.1	54.46	41.30	57.37	30.12	31.19	42.89
	Weight cluster	Uniform	8	3,837	19.0	55.56	41.13	57.53	29.61	31.06	**42.98**
	Router cluster	Uniform	8	3,837	19.0	55.25	41.39	57.41	29.61	31.16	42.96
	Anchor-based	Uniform	16	2,840	20.7	55.25	38.93	55.35	28.67	27.45	41.13
	Output cluster	Proportional	16	3,805	20.5	56.75	38.96	55.89	29.86	28.74	42.04
D-Optimal + ACP	Round-robin	Uniform	8	5,134	19.2	56.99	41.13	57.37	29.86	31.70	**43.41**
	Weight cluster	Proportional	16	4,333	20.6	54.22	39.59	54.67	28.24	28.91	41.13
	Router cluster	Uniform	8	5,134	19.2	56.59	40.99	57.45	29.18	32.30	43.30
	Anchor-based	Proportional	16	3,939	19.8	56.12	40.20	55.77	28.92	29.59	42.12
	Output cluster	Uniform	8	5,134	19.2	56.99	41.08	57.07	29.27	31.86	43.25

P Dense-to-Dense (D2D) pruning baseline

We implement a D2D baseline following the Minitron methodology [31]: structured pruning of a dense teacher (Qwen3-32B, 32B parameters) to a student of comparable size to our MoE-to-dense models, followed by distillation with the same dense teacher using matched hyperparameters and token budget.

Architecture search. Following the Minitron approach [31], we search over five candidate architectures at matched parameter count ($\sim$3.4B), varying pruning strategy (width-only vs. combined width+depth), number of layers, and hidden/FFN dimensions. All candidates are pruned using activation-based importance scoring [31] calibrated on 1,024 WikiText-103 samples and evaluated by WikiText-2 perplexity before distillation (Table 16).

Table 16: D2D architecture search: five candidate architectures pruned from Qwen3-32B (64 layers, d=5120, d_{dense}=25600). Width-only pruning that preserves all layers achieves the best pre-distill PPL. **Bold**: selected configuration.

Strategy	Layers	d	d_{dense}	Heads	Params	Pre-distill PPL
Width-only	**64**	**2,048**	**6,144**	**8**	**3.44B**	**15,300**
Width-only	64	1,536	8,192	12	3.29B	44,605
Combined	56	2,048	6,144	16	3.44B	20,299
Combined	48	2,048	6,144	16	3.44B	44,100
Combined	40	2,048	6,144	16	2.55B	286,698

The best architecture preserves all 64 layers with aggressive width pruning (d=2048, d_{dense}=6144, 3.44B parameters), achieving a pre-distill PPL of 15,300. Removing layers consistently degrades quality: even retaining 56 of 64 layers substantially increases PPL, suggesting that depth preservation is critical for structured pruning at this compression ratio.

Distillation. The selected D2D student is distilled with its dense teacher (Qwen3-32B) using identical hyperparameters, data, and token budget as all MoE-to-dense experiments. With this setup, D2D achieves only 33.28% average accuracy, barely above the random FFN baseline (32.70%, +0.6 pp) and far below even the weakest MoE-to-dense configuration (SF$\times$AB, 36.31%, +3.0 pp). This suggests that one-shot dense pruning at this compression ratio provides little structural advantage, while MoE-to-dense conversion preserves expert-level structure as a stronger initialization for distillation.

Iterative D2D pruning. Iterative pruning is the recommended practice for Minitron [31], so we also replace one-shot pruning with iterative prune-and-distill. Each schedule prunes the original Qwen3-32B down to $\sim$3.5B through repeated prune-and-distill stages (Table 17). The total token budget is held fixed across schedules and matched to all other 0.3B-token experiments. At each pruning stage, six candidate architectures are probed following Minitron, using a lightweight search over commonly used dimensions selected by perplexity. At every distillation stage, the original Qwen3-32B is the teacher.

Table 17: Iterative D2D schedules at a matched 0.3B-token budget. Iterative pruning improves on one-shot D2D by up to +6.9 pp, but DO-ACP leads the strongest schedule by +3.2 pp and on every benchmark.

Method	Ladder / token share	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
DO-ACP (ours)	–	**57.0**	**41.1**	**57.4**	**29.9**	**31.7**	**43.41**
Iterative D2D	8B$\rightarrow$5B$\rightarrow$3.5B, 17/33/50%	52.9	36.8	55.6	28.7	27.0	40.20
Iterative D2D	8B$\rightarrow$5B$\rightarrow$3.5B, 33/33/33%	52.6	36.6	55.0	28.0	27.0	39.84
Iterative D2D	8B$\rightarrow$6B$\rightarrow$4.5B$\rightarrow$3.5B, 25/25/25/25%	54.2	36.7	53.7	28.1	26.4	39.82
One-shot D2D	–	51.7	27.7	38.9	22.4	25.7	33.28

We also calibrate both methods on a corpus closer to the target distribution, built from the train splits of ARC, HellaSwag, MMLU, and Winogrande rendered as few-shot-style prompts. This corpus is

used for expert scoring in MoE-to-dense and for architecture selection in the three-stage iterative D2D baseline. It benefits MoE-to-dense more (+1.60 pp) than iterative D2D (+0.16 pp), widening the gap from 3.57 pp to 5.01 pp (Table 18), so calibration data does not explain the D2D deficit.

Table 18: MoE-to-dense and iterative D2D under WikiText and benchmark-train calibration (0.3B tokens).

Configuration	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
DO-ACP (WikiText)	57.0	41.1	57.4	29.9	31.7	43.41
DO-ACP (benchmark-train)	56.4	43.1	59.3	31.7	34.7	45.01
Iterative D2D (WikiText)	52.6	36.6	55.0	28.0	27.0	39.84
Iterative D2D (benchmark-train)	52.6	37.4	54.8	27.9	27.3	40.00

Scope of the comparison. The two teachers are comparable in total parameter count (30.5B vs. 32.8B) but not in depth or width: Qwen3-30B-A3B has 48 layers of width 2048, while Qwen3-32B has 64 layers of width 5120. Our student inherits the MoE teacher's geometry and attention layers, a structural advantage of MoE parents; the random FFN baseline (Table 3) shares this advantage, and with it held constant, expert selection contributes +10.7 pp.

Q Broader benchmark evaluation

All models in this section are base models. We therefore rely primarily on bits-per-byte (BPB), the prefill-only negative log-likelihood of held-out gold responses normalized by UTF-8 bytes, which stays graded at any capability level, together with few-shot and zero-shot accuracy where tasks remain discriminative.

Setup for Table 5. BPB corpora are HumanEval [6] and MBPP [2] reference solutions, GSM8K [8] and MATH [15] reference solutions, No Robots instruction responses, Glaive function-calling traces, and FLORES [14] passages in 10 languages, using up to 300 samples each at 2048 tokens. Math BPB is the mean over GSM8K and MATH. Accuracy metrics are GSM8K 5-shot exact match, Belebele [4] 0-shot over the same 10 languages, and RULER [17] at 4k context averaged over 11 self-contained subtasks (8 NIAH variants plus variable tracking, common-word extraction, and frequent-word extraction).

Science benchmarks. We also evaluate scientific knowledge on SciQ [42], OpenBookQA [30], and MMLU-STEM (Table 19). DO-ACP performs best among the students on every benchmark. MMLU-STEM shows the clearest separation, with DO-ACP +10.0 pp over SF and +13.3 pp over D2D.

Table 19: Science benchmarks after ∼4B-token distillation (accuracy, %).

Model	SciQ 0-shot	OpenBookQA 0-shot	MMLU-STEM 5-shot
DO-ACP	**93.6**	**40.2**	**40.1**
SF	93.0	39.2	30.1
D2D	92.0	37.0	26.8
Random FFN	88.7	33.8	28.4

R Calibration sensitivity

We measure how the top-8 experts selected per layer by DO-ACP respond to calibration size, sequence length, and domain. Overlap is the mean fraction of experts shared with a fixed reference configuration (WikiText, n=1024, maximum sequence length 2048), which is larger than the n=512 setting used for our main results; the two overlap by 98.4%. All domain corpora are packed to the same token count and length distribution as the reference, so only content differs (Tables 20 and 21). Across domains, 4.33 of the 8 experts per layer are selected by every domain, and 2.83 are selected by exactly one domain.

Table 20: Selection overlap with the reference calibration configuration.

Axis	Configuration	n	Seq.	Tokens	Overlap	Per-layer min
Size	4× smaller	256	2048	43,414	91.7%	62%
Size	2× smaller (main results)	512	2048	85,352	98.4%	88%
Size	2× larger	2048	2048	320,590	95.3%	88%
Seq. length	Half	1024	1024	174,956	96.4%	88%
Seq. length	Double	1024	4096	174,956	97.1%	88%
Domain	Code	1024	2048	174,998	70.3%	38%
Domain	Benchmark-mix	1024	2048	174,298	73.4%	50%
Domain	Math	1024	2048	174,990	75.0%	50%
Domain	Instruction	1024	2048	174,978	78.4%	38%

Table 21: Pairwise selection overlap between calibration domains (%).

	WikiText	Code	Math	Instruction	Benchmark-mix
WikiText	100.0	70.3	75.0	78.4	73.4
Code	70.3	100.0	75.5	75.5	69.0
Math	75.0	75.5	100.0	82.3	77.1
Instruction	78.4	75.5	82.3	100.0	78.6
Benchmark-mix	73.4	69.0	77.1	78.6	100.0

To test whether these shifts matter downstream, we re-run the pipeline with the shifted selections (Table 22). A 25–27% change in selected experts moves final accuracy by only ∼1–1.6 pp, leaving our conclusions unchanged.

Table 22: Downstream accuracy after re-running the pipeline with shifted selections (0.3B tokens).

Calibration	Overlap	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
WikiText (main results)	–	57.0	41.1	57.4	29.9	31.7	43.41
Math	75.0%	56.3	42.1	59.1	31.2	33.4	44.42
Benchmark-mix	73.4%	56.4	43.1	59.3	31.7	34.7	45.01

S Deployment measurements

We measure resident memory, peak VRAM, throughput, and latency on a single H200 GPU, using the vLLM engine [21] for throughput and latency. TTFT is the time to first token (prefill) and TPOT is the time per output token (decode). b denotes batch size and s prompt length. The dense student has lower weight memory, higher throughput, and faster decode than its MoE parent in every setting measured (Tables 23 and 24).

Table 23: Weight memory and peak VRAM.

				Peak VRAM (GiB)		
Model	Params	Weights (GiB)	vs. ours	$b1, s128$	$b32, s2048$	$b32, s4096$
MoE teacher (Qwen3-30B-A3B)	30.532B	56.9	9.14×	56.9	66.0	75.2
Dense student (ours)	3.340B	6.2	1.00×	6.3	15.5	24.8
D2D teacher (Qwen3-32B)	32.762B	61.0	9.81×	61.1	88.9	116.8
D2D student	3.441B	6.4	1.03×	6.5	21.7	36.9

Table 24: Throughput (tokens/s) by batch size and latency at batch 32, by prompt length.

Prompt	Model	$b1$	$b4$	$b16$	$b32$	TTFT (ms)	TPOT (ms)
128	MoE teacher	217	686	1,618	2,712	80	11.26
	Dense student (ours)	279	1,078	4,067	7,501	51	3.90
	D2D teacher	54	210	790	1,416	410	19.55
	D2D student	272	1,056	3,942	6,832	49	4.33
2048	MoE teacher	202	584	1,226	1,752	1,000	10.53
	Dense student (ours)	261	879	2,155	2,893	742	5.31
	D2D teacher	50	157	349	433	6,524	23.12
	D2D student	252	818	1,914	2,468	769	7.01
4096	MoE teacher	190	494	955	1,118	2,160	11.83
	Dense student (ours)	242	713	1,372	1,630	1,653	6.77
	D2D teacher	46	122	212	241	13,607	26.92
	D2D student	233	652	1,193	1,377	1,723	9.86

T Model architecture details

Table 25 summarizes the original teacher architectures and the dense students produced by our pipeline for all three models. Below, for each non-Qwen3 model we describe the original MoE architecture and then the adjustments needed to convert it to a dense FFN whose width matches the teacher's active FFN computation per token.

Table 25: Architecture of each MoE teacher and the corresponding dense student produced by our pipeline. †DeepSeek-V2-Lite has 2 shared experts per MoE layer; layer 0 is a standard dense FFN (zero-padded to match d_{dense}).

Property	Qwen3-30B-A3B	DeepSeek-V2-Lite	GPT-OSS-20B
Teacher			
Total parameters	30.5B	15.7B	20.9B
Active parameters	3.3B	2.4B	3.6B
Hidden dimension (d)	2,048	2,048	2,880
Routed experts / layer (E)	128	64†	32
Active routed experts (k)	8	6	4
Shared experts / layer	–	2†	–
Dense Student			
Total / active parameters	3.3B	2.4B	3.6B
Groups	8	8 (2 shared + 6 routed)	4
FFN intermediate dim (d_{dense})	6,144 ($= 8\times768$)	11,264 ($= 8\times1{,}408$)	11,520 ($= 4\times2{,}880$)

DeepSeek-V2-Lite. The original 16B-parameter model has 64 routed experts plus 2 shared (always-on) experts per MoE layer with top-k=6 routing, expert intermediate dimension $1{,}408$, and a single non-MoE first layer (a standard dense FFN of intermediate dimension $10{,}944$). Unlike Qwen3 and GPT-OSS, it does *not* renormalize routing probabilities after the top-k cut, so the routed experts' weights do not sum to one. To convert, we treat the two shared experts as always-on additions alongside the routed experts: $2 + 6 = 8$ FFN components are active per token, giving $d_{\text{dense}} = 8 \times 1408 = 11{,}264$. The shared experts are copied directly with DP scale 1.0 since router weighting does not apply to them, and our K parameter governs only the routed experts. Because routing is not renormalized, we use each group's average conditional probability as the DP scaling ratio, matching the average scaling the MoE applies to each expert's output. The first layer is zero-padded from $10{,}944$ to $11{,}264$ to match subsequent layers.

GPT-OSS-20B. The original 21B-parameter model has 32 routed experts per MoE layer with top-k=4 routing, expert intermediate dimension $2{,}880$, and renormalizes routing probabilities after the top-k cut. It is a post-trained reasoning model that generates chain-of-thought traces before answers

in its native chat-template format. To convert, we set $d_{\text{dense}} = 4 \times 2880 = 11{,}520$. Because routing is renormalized (matching Qwen3), we sweep both uniform and proportional DP scaling (Section 2.4). We evaluate all models (teacher and students) in standard completion mode for consistency with the other two architectures; this underestimates the teacher's native-format capability (e.g., MMLU 49% in completion mode vs. 72% with the chat template).

U Cross-model full results

Tables 26 and 27 provide per-benchmark results for all cross-model configurations.

Table 26: DeepSeek-V2-Lite cross-model validation (16B → dense, 0.3B-token distillation). Rows sorted by scoring method; both K=6 (pure pruning) and K=12 (merging) shown. **Bold**: best per column.

Scoring	K	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
SF	6	54.9	38.9	52.4	27.1	26.9	40.04
SF	12	56.9	41.1	**54.8**	**28.4**	24.6	41.16
CP	6	53.0	36.9	49.4	25.7	25.3	38.07
CP	12	55.6	40.2	53.5	26.6	26.8	40.53
ACP	6	56.8	38.6	51.0	27.5	28.1	40.37
ACP	12	57.1	40.9	52.9	27.4	26.4	40.93
DO-ACP	6	**60.3**	41.0	53.7	28.2	**28.7**	**42.39**
DO-ACP	12	59.0	**41.5**	51.7	26.2	26.9	41.07
Random FFN + teacher attn		50.6	25.6	30.6	20.9	23.6	30.25
Random initialization		50.1	25.4	28.8	24.1	22.9	30.27
Teacher (DeepSeek-V2-Lite)		76.2	80.5	84.4	56.3	58.0	71.09

Table 27: GPT-OSS-20B cross-model validation (21B → dense, 0.3B-token distillation). Both K=4 (pure pruning) and K=8 (merging) shown. **Bold**: best per column. †Post-trained reasoning model evaluated in completion mode; native-format performance is expected to be substantially higher.

Scoring	K	Wino	Hella	ARC-E	ARC-C	MMLU	Avg (%)
SF	4	51.6	29.4	34.0	22.4	23.3	32.15
SF	8	51.6	29.3	33.3	21.7	22.8	31.72
CP	4	50.9	30.4	**36.4**	23.0	23.6	32.86
CP	8	49.7	29.3	32.2	23.2	23.1	31.49
ACP	4	53.0	31.9	35.6	23.0	23.3	33.36
ACP	8	**53.1**	30.5	33.6	23.3	**23.7**	32.82
DO-ACP	4	53.0	**32.1**	36.7	23.2	**23.7**	**33.71**
DO-ACP	8	51.3	29.9	33.5	22.5	23.3	32.11
Random FFN + teacher attn		50.2	27.5	28.5	23.0	23.2	30.46
Random initialization		50.0	26.0	25.8	**25.3**	23.0	30.02
Teacher (GPT-OSS-20B)†		59.3	39.9	80.9	53.7	49.6	56.67

V Compatibility with modularity-aware pretraining

We have so far evaluated our pipeline on standard MoE teachers. An approach complementary to ours, introduced by Wang et al. [41], enhances the compressibility of MoE models during pretraining by constraining each training document to route through a sampled per-document expert pool. The resulting model retains downstream performance under expert subset pruning. While the original focus of the method is on expert-pruned models that retain the router structure, we investigate its compatibility with our pipeline, which produces a fully dense student.

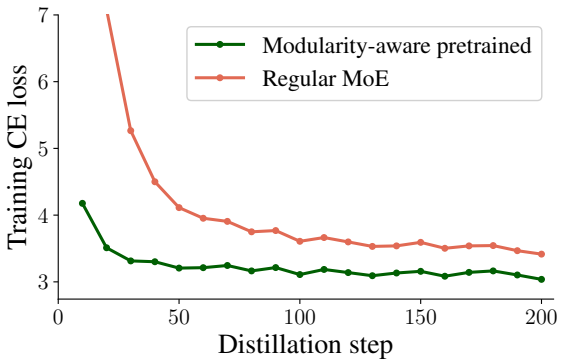


Figure 8: Cross-entropy on the training data during MoE-to-dense distillation (DO-ACP, $K{=}k$, forward-KL objective) from matched modularity-aware pretrained and regular MoE teachers. The modularity-aware student remains lower throughout.

We experiment with the open-source weights of Wang et al. [41]: a 14B-parameter MoE with $E{=}127$ routed experts plus 1 shared expert and $k{=}7$ (1B active). As a control, we use the matched regularly pretrained MoE released alongside, sharing architecture and training data. We apply our strongest configuration (DO-ACP scoring with pure pruning at $K{=}k{=}7$ routed experts plus the model's single shared expert) to both teachers, yielding two 1.5B dense students. Distillation follows the same protocol as Section 3.1.

Table 28 reports downstream accuracy and WikiText-2 perplexity before and after distillation. The modularity-aware pretrained teacher yields a $+3.61$ pp downstream lift and an $\sim 87\times$ lower pre-distill PPL relative to the regular MoE control. Figure 8 shows that this initialization advantage persists throughout distillation: the student distilled from the modularity-aware teacher begins at substantially lower training CE and remains below its regular MoE counterpart. We view this as preliminary evidence that compression-aware pretraining could pair well with our MoE-to-dense pipeline, and leave a fuller co-design of the two stages to future work.

Table 28: Compatibility with modularity-aware pretraining (DO-ACP, $K{=}k{=}7$). The modularity-aware pretrained teacher yields both lower pre-distill PPL and higher downstream accuracy than the regular MoE control. *Pre*/*Post* = WikiText-2 PPL before/after distillation.

	PPL (↓)		**Downstream Accuracy (%)**					
Teacher	**Pre**	**Post**	**Wino**	**Hella**	**ARC-E**	**ARC-C**	**MMLU**	**Avg**
Regular MoE	13,549	31.13	52.6	33.8	44.5	23.0	24.3	35.67
Modularity-aware pretrained	**156**	**19.76**	**56.9**	**40.1**	**49.1**	**25.1**	**25.2**	**39.28**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state our contributions (DO-ACP scoring, MoE-to-dense pipeline, +6.3 pp over Dense-to-dense pruning at ∼4B-token distillation, cross-model generalization), each demonstrated in Sections 3.2–3.5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 5, covering merge-weight coupling, training-budget scope, competitiveness with off-the-shelf models, the one-shot D2D baseline at extended budget, domain-specific capability, and the cross-model expert-pool trend.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theorem 2.1 (main text), Proposition F.1, and Theorem F.3 are stated with assumptions and proved in Appendix F; additional theoretical results (finite-sample calibration, grouping, merging) are proved in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental setup is in Section 3.1; full hyperparameters and infrastructure are in Appendix M; distillation methodology and ablations in Appendix J; per-architecture conversion details in Appendix T.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/krafton-ai/moe-to-dense`. The methodology, full hyperparameters, calibration and distillation data sources, and per-architecture conversion details are described in the paper and appendices to enable reimplementation; all teacher models and evaluation benchmarks are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.1 specifies models, calibration data, distillation data, batch size, sequence length, optimizer, and learning-rate schedule; full hyperparameters and infrastructure are in Appendix M; benchmark tasks and few-shot configurations are listed under "Evaluation" in Section 3.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Each cell of the 350-configuration merge sweep and each distillation run is a single deterministic seed; multi-seed runs at this scale were prohibitively expensive in compute. Robustness is instead checked via cross-architecture validation (DeepSeek-V2-Lite, GPT-OSS-20B in Section 3.5) and a base-vs-instruct teacher replication (Appendix L).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware and per-run time are reported in Appendix M (4×H200 GPUs; ∼5.5 h per 200-step distillation), and the wall-clock comparison in Section 3.3 reports the 2×B200-GPU setup. The full project also required preliminary runs not reported in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses publicly released models and datasets, involves no human subjects, and complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper develops a model-compression methodology applied to publicly available open-weight models. Reducing memory footprint broadens deployment access; we do not foresee specific negative societal impacts beyond those associated with general-purpose language models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release a new model. The compressed dense student is at most as capable as its publicly released open-weight teacher and presents no additional misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All teacher models (Qwen3-30B-A3B, DeepSeek-V2-Lite, GPT-OSS-20B), evaluation benchmarks (Winogrande, HellaSwag, ARC-Easy/Challenge, MMLU, and the additional benchmarks in Appendices Q and R), and training/calibration corpora (FineWeb-Edu, WikiText) are cited or named at first use, and we use the publicly released versions under their respective open licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code with documentation at `https://github.com/krafton-ai/moe-to-dense`; we do not release new models or datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.