RDO-MoE: Rate-Distortion Optimized Joint Rank-Sparsity Allocation for Mixture-of-Experts Compression

Anonymous authors
Paper under double-blind review

Abstract

Sparse mixture-of-experts (MoE) language models store far more parameters than any one token activates; whether one runs on a commodity accelerator therefore hinges on how its expert bank is compressed, not on how sparsely it is routed. Existing pipelines settle that representation in stages, pruning before rating the survivors or scoring the base and the residual ranks under separate objectives, and can thereby miss the configurations favorable only when both are set together. We propose **RDO-MoE** (Rate–Distortion Optimized MoE compression), which prices shared-base sparsity and expert-residual rank together on one rate–distortion surface under a global serialized-byte budget. Joint per-layer surfaces turn each layer's weights and routing statistics into one finite table of nondominated pairs. Multiplier-indexed allocation dualizes that budget into one Lagrange multiplier and bisects it for a feasible integer allocation over the per-layer frontiers. A capability diagnostic outside the reconstruction objective reports signed per-task differences, not one score. The key insight is that the base and the rank index one surface rather than two budgets; pricing both against a single multiplier thus turns an ordered pair of searches into one query. On Qwen1.5-MoE-A2.7B at 40% compression it reaches 8.91 perplexity and a 55.7 aggregate against 9.16 and 55.4 for the best prior method; on Mixtral-8×7B it reaches 5.49 against 5.88 at the same budget and the lowest perplexity at every budget. On the instance exact search can enumerate it stays within 0.0057 in relative perplexity at 19.6× the solver speed, while the 0.9- and 1.3-point WinoGrande and BoolQ regressions at 40% stay visible rather than being averaged away by the nine-task aggregate.

1 Introduction

Sparse mixture-of-experts (MoE) language models scale capacity by routing each token to a few experts out of many (Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024). The design has scaled from the first sparsely gated layer to the systems in service today (Shazeer et al., 2017; Lepikhin et al., 2021; Du et al., 2022; Rajbhandari et al., 2022), and the gap between the parameters a token activates and the parameters a checkpoint must hold is exactly what makes it attractive to train and awkward to serve. What that sparsity saves is arithmetic: the experts a token skips contribute no multiplications to its forward pass. It saves no bytes: every expert stays resident in storage and in device memory whether or not it is visited. Deployment is therefore bounded by the checkpoint rather than by the forward pass, and whether a model can be served at all is settled by whether its bytes fit.

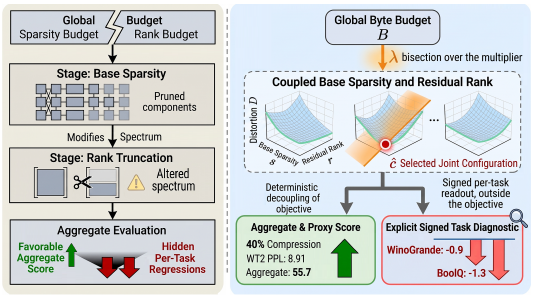


Figure 1: **Left:** staged pipelines fix the base and the ranks in separate steps, and one aggregate hides per-task regressions. **Right:** RDO-MoE prices both on one surface under one budget and reports signed per-task movement.

Three families of compression act on those bytes. Expert pruning removes whole experts or expert groups, shrinking the parameter pool itself (Lasby et al., 2025; Liu et al., 2026; Yang et al., 2024).

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

Low-rank methods keep every expert in place and instead reduce the dimension of each weight matrix, with the saving coming from the spectrum rather than from the component set (Yuan et al., 2023; Wang et al., 2025c;b; Hu et al., 2026; Mi et al., 2025; 2026; Li et al., 2025). Shared-base decompositions combine the two views, storing the structure common to the experts once and encoding each expert as a residual against it (Gu et al., 2025). Together these lines have brought MoE checkpoints within the memory of a single accelerator. Each of them fixes the shape of the representation in stages.

That staging is a structural commitment: every one of the three families decides which components survive before deciding how finely what survives is encoded. Three consequences follow. The rank decision meets a spectrum the base decision has already reshaped; the layers settle their shares of the budget with nothing aligning both decisions against one global byte constraint, and the only route that aligns them exactly can cost multiplicatively in depth; and the quality of an allocation is read off a single aggregate that averages away the tasks that moved the wrong way. Figure 1 contrasts the two decision orders and the two ways of reporting what each costs. These three are coupled: the base fixes the spectrum, and the aggregate then hides what that costs. Shared-base sparsity and residual rank are therefore two faces of one decision, and an order that fixes either first can miss the configurations that pay only when both are set together.

To address these coupled challenges, we propose RDO-MoE (Rate–Distortion Optimized MoE compression), a joint allocator that treats shared-base sparsity and expert-residual rank as one budgeted decision. RDO-MoE comprises three components: (1) a per-layer joint surface that enumerates sparsity and rank in pairs, with rank truncation acting on the spectrum the base decision induces; (2) a multiplier-indexed allocation that bisects one multiplier against the global serialized-byte budget, sweeping the lower convex envelope of every per-layer frontier with feasibility and slack checks; and (3) a signed capability diagnostic held outside the reconstruction objective, reporting direction-normalized per-task differences instead of collapsing them into one score. The key insight is that the base and the rank index one surface rather than two budgets, so pricing both against a single multiplier turns an ordered pair of searches into one query. Our contributions are as follows:

- We identify the joint-allocation gap in MoE compression: no existing method prices shared-base sparsity and residual rank against each other under one byte budget, so a rank truncation meets a spectrum chosen without it and one aggregate hides per-task regressions.
- We propose RDO-MoE, a joint rate–distortion allocator that enumerates every layer as one surface over sparsity and rank, bisects a single multiplier against the global serialized-byte budget, and keeps a signed per-task diagnostic outside the reconstruction objective.
- Experiments on Qwen reach 8.91 perplexity and a 55.7 aggregate score at 40% compression against 9.16 and 55.4 for D-Rank; on Mixtral RDO-MoE holds the lowest perplexity at every budget; and the diagnostic reports the 0.9 and 1.3 point regressions on WinoGrande and BoolQ at 40% that the aggregate hides.

2 MOTIVATION

To understand why staged compression misallocates the budget under one serialized-byte constraint, we examined how an allocation is stated, solved, and read, revealing three challenges.

Challenge #1: Spectrum drift under staging. We compared two published staged pipelines that fix opposite halves of the decision first, a delta decomposition that fixes the shared base and a rank-only allocation that fixes the rank. Figure 2 (left) shows that they do not agree at any ratio and that they agree least where the budget is tightest, 0.69% apart at 30% compression against 1.47% at 60%. This drift has a structural source: a shared-base decomposition fixes a common expert structure first and encodes expert-specific residuals against it, which places the base inside every residual spectrum a later rank truncation must act on. Staged pipelines can therefore fail by construction: the base is selected under an objective that cannot see the ranks it will induce, so the configurations that pay only when both are set together can be discarded before the rank decision exists. *How should base sparsity and residual rank be chosen as one decision rather than as two independent ones?*

Table 1: Exact dynamic programming (DP) in perplexity (PPL).

Search	PPL	Time (s)	Speed
Exact DP	8.82	159.4	1.0×
EvoESAP	8.93	39.4	4.0×

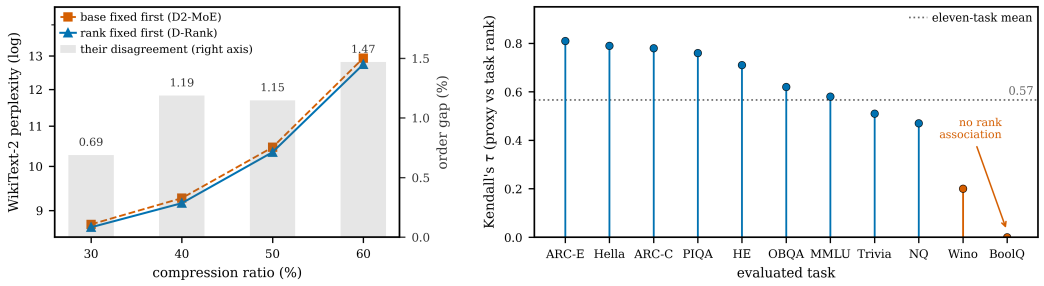


Figure 2: **Left:** two staged orders disagree at every ratio, most where the budget is tightest. **Right:** the proxy tracks task rank unevenly, and on one task not at all. More details in Appendix A.2.

Challenge #2: Multiplicative search cost. We timed an exact search over the per-layer surfaces and one published heuristic search on the bounded instance used for the exact comparison, a Qwen instance admitted only because the nondominated partial states the exact route must retain stay under a fixed cap. Table 1 reports them: exact DP takes 159.4 seconds and EvoESAP 39.4, a $4.0\times$ difference at a relative perplexity-space gap of 0.0125. The obstacle is structural: once the two decisions are coupled, a layer that sets its own share in isolation cannot reach the reallocations that cross layers. The one route that does coordinate them exactly sweeps a state space that can grow with the product of the per-layer grids, a cost that can multiply with depth rather than add to it. *How can L layers be coordinated against one global budget at a cost that stays near-linear in depth?*

Challenge #3: A sign-blind proxy. We ranked configurations by reconstruction error and by task performance, and measured their agreement on the eleven evaluated tasks. Figure 2 (right) reports a mean Kendall's τ of 0.57 spanning 0.81 on ARC-Easy down to 0.20 on WinoGrande and 0.00 on BoolQ, so the agreement runs from strong to none, and on the weakest task the ranking the proxy induces has no rank association with the task ranking. A squared-error objective is sign-blind by construction, because it is invariant to which task a residual direction serves; an improvement in the proxy therefore guarantees no sign for any single task, and averaging restores no such guarantee. *How can taskwise movement be read out with its signs intact instead of being collapsed into the reconstruction objective?*

3 Problem Formulation

To address the challenges of Section 2, we model MoE compression as one budgeted allocation over per-layer configurations,

$$\min_{c_1,\dots,c_L} \sum_{l=1}^{L} D_l(c_l) \quad \text{s.t.} \quad \sum_{l=1}^{L} R_l(c_l) \le B, \qquad c_l \in \mathcal{C}_l, \tag{1}$$

where $c_l \in \mathcal{C}_l$ pairs a shared-base sparsity with the ranks retained for the residuals it induces, D_l is the routing-weighted reconstruction error it leaves in layer l, R_l its serialized storage under one fixed representation, and B the global byte budget.

Challenge #1 is carried by the distortion D_l: the sparsity in c_l fixes the spectrum its ranks must truncate. Challenge #2 is carried by the single budget, the only tie between L otherwise independent layers. Challenge #3 is carried by what Eq. 1 leaves out, since task behavior never enters D_l and is read out separately. More details are provided in Appendix A.3.

4 RDO-MoE

4.1 Overall Framework

To resolve the problem mentioned in Section 3, we propose RDO-MoE, a framework of three components, illustrated in Figure 3. Joint per-layer surfaces turn a layer's fixed weights and routing statistics into one finite table of nondominated rate–distortion pairs. Multiplier-indexed allocation queries those tables with one scalar under the global byte budget and returns a budget-feasible configuration. A capability diagnostic reads signed taskwise movement outside the objective and returns a per-task report. End-to-end integration runs the three in one pass.

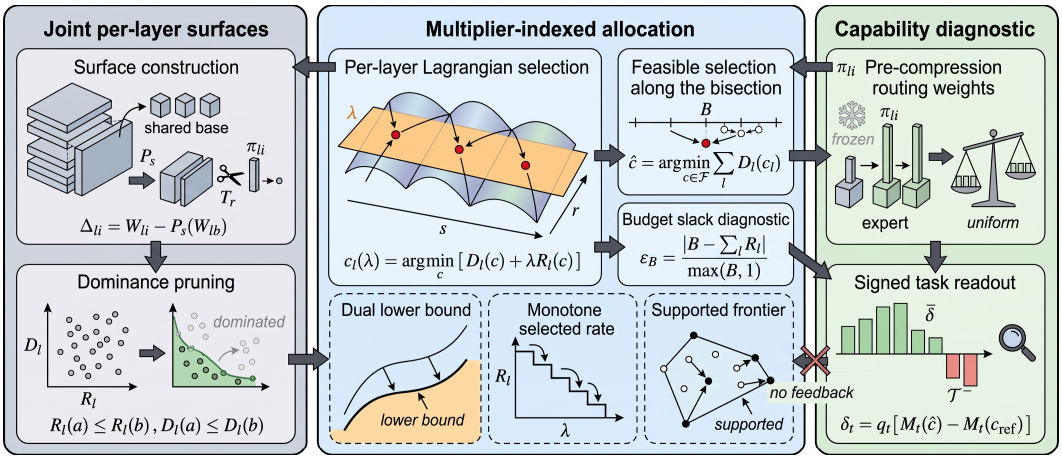


Figure 3: **Left:** each layer is enumerated into rate–distortion pairs, then pruned. **Middle:** one multiplier prices all layers, bisected against the byte budget. **Right:** routing weights frozen before selection, signed task movement read after. **Bottom:** the properties that make the search well posed.

Operationally, RDO-MoE prunes each layer's shared base at every grid sparsity, recomputes every expert residual against it, truncates the residual at every grid rank, and records the bytes and routing-weighted error of each pair. It drops dominated pairs and selects one pair per layer with one Lagrange multiplier. Since each rank is scored on the spectrum its base sparsity leaves, every query prices the base–residual interaction.

4.2 Joint per-layer surfaces

To address Challenge #1, the spectrum drift, we propose joint per-layer surfaces. As illustrated in Figure 3 (left), this component runs once per layer: it enumerates every admitted sparsity–rank pair, scores each as one rate–distortion point, and then prunes the dominated ones.

Pre-compression routing weights. The weights that score the surface are fixed before any configuration is selected: π_{li} are the routing frequencies of expert i in layer l, measured once on the calibration subset, nonnegative and summing to one over the E_l experts of that layer. Uniform weights form a control, while held-out routing is used only as a diagnostic.

Surface construction. Since the base choice reshapes every residual spectrum, the surface construction evaluates the shared-base and residual-rank decisions jointly. The enumeration ties all experts in a layer to a common rank, so each configuration reduces to a pair over the sparsity grid $\mathcal{S}$ and a shared residual-rank grid $\mathcal{R}_0$, and the per-layer count $|\mathcal{C}| = |\mathcal{S}| \times |\mathcal{R}_0|$ is common to all layers. Let W_{li} denote the expert matrices of layer l, W_{lb} their shared base, and P_s a deterministic base-compression operator indexed by sparsity $s \in \mathcal{S}$. The shared base is a fixed input; its extraction lies outside the allocation. For each $c_l \in \mathcal{C}_l$, RDO-MoE applies P_s and recomputes the expert residual

$$\Delta_{li}(s) = W_{li} - P_s(W_{lb}), \tag{2}$$

truncates it with a best rank-r approximation T_r in Frobenius norm, and records the pair $(R_l(c_l), D_l(c_l))$. The distortion scored on the surface is

$$D_l(c_l) = \sum_{i=1}^{E_l} \pi_{li} \, \| W_{li} - P_{s_l}(W_{lb}) - T_{r_{li}}(\Delta_{li}(s_l)) \|_F^2 . \tag{3}$$

Eq. 3 is written for the general configuration $c_l = (s_l, r_{l1}, \ldots, r_{lE_l})$ and specializes to $r_{li} = r_l$ under the shared per-layer rank enumerated here, which is what keeps the admitted set constant in E_l rather than exponential.

Dominance pruning. A configuration a dominates b when

$$R_l(a) \le R_l(b), \qquad D_l(a) \le D_l(b), \tag{4}$$

with at least one strict inequality. Removing dominated configurations does not change the optimum of Eq. 1, and we write $\mathcal{P}_l$ for the frontier retained in layer l. The retained frontier preserves the joint degrees of freedom on which the base–residual interaction acts.

4.3 MULTIPLIER-INDEXED ALLOCATION

To address Challenge #2, the multiplicative search cost, we propose a multiplier-indexed allocation. As illustrated in Figure 3 (middle), it prices every layer with one scalar, selecting a configuration at each multiplier, bisecting that scalar against the global budget, and returning the best feasible one.

Per-layer Lagrangian selection. A common multiplier coordinates the layerwise queries under the single global byte constraint. For $\lambda \geq 0$, each layer selects on its retained frontier

$$c_l(\lambda) \in \arg\min_{c \in \mathcal{P}_l} \big[D_l(c) + \lambda R_l(c)\big], \tag{5}$$

with ties broken toward lower serialized rate, then the enumeration index. Selecting on $\mathcal{P}_l$ loses nothing: for $\lambda \geq 0$, a dominated configuration never beats the one dominating it.

Dual lower bound. The same multiplier admits a dual reading:

$$g(\lambda) = \sum_l \min_{c \in \mathcal{P}_l} \big[D_l(c) + \lambda R_l(c)\big] - \lambda B. \tag{6}$$

It is a lower bound on Eq. 1.

Monotone selected rate. Selecting the per-layer minimizer of Eq. 5 at each multiplier makes the total selected rate $\sum_l R_l(c_l(\lambda))$ non-increasing in λ: a larger multiplier never selects a higher total rate. This monotonicity is what makes the bisection search over λ well posed. More details about the exchange argument behind Proposition S2 are provided in Appendix A.3.

Supported frontier. We call a configuration supported if some $\lambda \geq 0$ selects it through Eq. 5, which places it on the lower convex envelope of the per-layer frontier; a multiplier sweep reaches only supported configurations, which is what a supported-frontier solution means throughout.

Feasible selection along the bisection. If $B < \sum_l \min_{c \in \mathcal{P}_l} R_l(c)$, the procedure reports the budget as infeasible and returns no configuration. Otherwise, the minimum-rate configuration is feasible and the multiplier path has at least one feasible member. If the selection at $\lambda = 0$ already fits, it is returned at once. Otherwise the search brackets the budget: a trial upper multiplier is doubled until the rate it selects fits, and each step tests the midpoint and keeps the half whose endpoints still bracket the budget, until the relative bracket width falls below the requested tolerance. The allocator collects the feasible configurations produced at the visited multipliers,

$$\mathcal{F} = \Big\{ c(\lambda_j) : \textstyle\sum_l R_l(c_l(\lambda_j)) \leq B,\ \lambda_j \text{ visited along the bisection} \Big\},$$

and returns the lowest-distortion member of that set,

$$\hat{c} = \arg\min_{c \in \mathcal{F}} \sum_l D_l(c_l). \tag{7}$$

Budget slack diagnostic. The normalized unused-budget diagnostic is

$$\epsilon_B = \frac{|B - \sum_l R_l(\hat{c}_l)|}{\max(B, 1)}. \tag{8}$$

Since $\hat{c}$ is feasible by construction, ϵ_B in Eq. 8 measures the unspent budget. A common multiplier can skip unsupported discrete optima; it returns a point on the supported frontier; exact DP, the comparison oracle, computes the optimal distortion $D^\star(B)$ of Eq. 1 on enumerable instances.

4.4 CAPABILITY DIAGNOSTIC

To address Challenge #3, the sign-blind proxy, we propose a capability diagnostic held outside the objective. As illustrated in Figure 3 (right), this component reads direction-normalized per-task differences after selection, against a reference evaluated under the same protocol.

Signed task readout. For a downstream metric M_t, let $q_t = +1$ when larger is better and $q_t = -1$ when smaller is better. The capability check reports the direction-normalized paired difference

$$\delta_t = q_t \big[M_t(\hat{c}) - M_t(c_{\mathrm{ref}}) \big] \tag{9}$$

against a fixed reference c_{ref}, the compared allocation at the same budget, named with each reported comparison and evaluated under an unchanged template, split, and aggregation rule. The signs of Eq. 9 give two readouts, one aggregate over a family $\mathcal{G}$ of commensurable metrics and one exception set over all evaluated tasks $\mathcal{T}$,

$$\bar{\delta} = \frac{1}{|\mathcal{G}|} \sum_{t \in \mathcal{G}} \delta_t, \qquad \mathcal{T}^- = \{ t \in \mathcal{T} : \delta_t < 0 \}. \tag{10}$$

End-to-end integration. Given a checkpoint and one global byte budget, RDO-MoE compresses the model in a single pass. As illustrated in Figure 3, the joint surface is built once per layer over the fixed finite grids and pruned to its nondominated points; the allocator then bisects one multiplier against the budget, a search kept well posed by the monotone selected rate of Proposition S2, and returns the best feasible configuration it visits. That configuration is realized in one pass and serialized under the same byte convention used to construct R_l, and the capability diagnostic reads the resulting model against the reference. The output is one serialized checkpoint together with its per-task readout. More details are provided in Appendix A.4.

5 Experiments

5.1 Implementation Details

Software and hardware. All runs use an NVIDIA A100 80 GB GPU, CUDA 12.8, and PyTorch 2.11; weights and the forward pass are bfloat16, and the residual factorization is float32. Model releases and the harness version are pinned. More details are provided in Appendix A.5.

Datasets. WikiText-2 (Merity et al., 2017) is the perplexity and calibration corpus. Capability covers ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), BoolQ (Clark et al., 2019), OpenBookQA (Mihaylov et al., 2018), MMLU (Hendrycks et al., 2021), TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and HumanEval (Chen et al., 2021). Details in Appendix A.6.

Backbones. The primary backbone is Qwen1.5-MoE-A2.7B (Qwen Team, 2024) with $L = 24$ layers; Mixtral-8×7B (Jiang et al., 2024) with $L = 32$ supplies the cross-backbone comparison. Both span 30%, 40%, 50%, and 60% compression, measured as reduction from the serialized representation, so all methods at one point share the same byte convention. Details in Appendix A.7.

Baselines. Uniform+Magnitude is the simple storage control. SAES-SVD and D-Rank represent low-rank allocation, D2-MoE represents delta decomposition, and EvoESAP represents nonuniform expert pruning. Exact dynamic programming is admitted only on the bounded solver instance, and a greedy allocator is the simple-allocator control. Details in Appendix A.8.

Evaluation metrics. Lower perplexity and higher task score are better. The aggregate is the unweighted mean of the nine accuracy-scored tasks; Natural Questions and HumanEval are reported separately, since exact match and pass@1 are not commensurable with accuracy. Resource accounting covers construction and allocation time, memory, and throughput. Details in Appendix A.9.

Hyperparameters. The grids hold $|\mathcal{S}| = 7$ base sparsities and $|\mathcal{R}_0| = 8$ residual ranks, giving 56 configurations per layer under one rank per layer. The base operator prunes by magnitude with row-major ties; equal objectives prefer lower serialized rate, then the enumeration index; the multiplier search stops at relative tolerance 10^{-3}. More details are provided in Appendix A.10.

5.2 Main Results

Compression quality. Table 2 reports the principal model–method–compression comparison. On Qwen, RDO-MoE has the lowest perplexity and the highest aggregate score at all four budgets:

Table 2: Compression quality across backbones, methods, and budgets. Lower PPL and higher Score are better; bold marks the best per column within each backbone block.

		30%		40%		50%		60%	
Model	Method	PPL	Score	PPL	Score	PPL	Score	PPL	Score
Qwen	Uniform+Magnitude	9.71	54.7	10.84	52.9	13.06	50.1	17.84	46.2
Qwen	SAES-SVD	8.95	56.0	9.62	54.6	11.18	52.1	14.35	48.4
Qwen	EvoESAP	8.78	56.5	9.31	55.1	10.62	52.7	13.26	49.0
Qwen	D-Rank	8.65	56.8	9.16	55.4	10.34	53.0	12.74	49.4
Qwen	D2-MoE	8.71	56.7	9.27	55.2	10.46	52.8	12.93	49.2
Qwen	**RDO-MoE**	**8.51**	**57.0**	**8.91**	**55.7**	**9.94**	**53.2**	**12.05**	**49.7**
Mixtral	EvoESAP	5.76	64.9	5.96	63.3	6.47	60.6	7.40	57.4
Mixtral	D-Rank	5.68	64.9	5.88	63.6	6.33	**60.9**	7.18	57.7
Mixtral	D2-MoE	6.00	64.8	6.22	63.3	6.70	60.4	7.62	57.4
Mixtral	**RDO-MoE**	**5.33**	**65.2**	**5.49**	**64.1**	**5.86**	60.8	**6.58**	**58.3**

at 40% its 8.91 perplexity improves on D-Rank (9.16), D2-MoE (9.27), EvoESAP (9.31), SAES-SVD (9.62), and Uniform+Magnitude (10.84), with an aggregate of 55.7 against 55.4, 55.2, 55.1, 54.6, and 52.9; at 60% the margin widens to 12.05 against 12.74 for D-Rank and 17.84 for Uniform+Magnitude. Mixtral-8×7B keeps the joint allocator lowest on perplexity on a second backbone, 5.33 against 5.68 at 30% and 6.58 against 7.18 at 60%, while its task aggregate holds the four methods within one point. The aggregate moves less than perplexity partly because it nets task gains against task losses, which the signed readings below separate.

Budget progression. Figure 4 plots Qwen WikiText-2 perplexity against compression ratio on a log axis, where lower is better. RDO-MoE is lowest at every ratio, and its margin over D-Rank widens from 0.14 at 30% to 0.69 at 60% (1.6% to 5.4%); even the smallest margin is ten times the ± 0.014 resampling half-width measured at 40%. The gap to the stage-wise D2-MoE grows faster, from 0.20 to 0.88, consistent with Challenge #1: heavier base sparsity reshapes more of the residual spectrum that a rank decision fixed afterwards must meet, and staging costs more as the budget tightens.

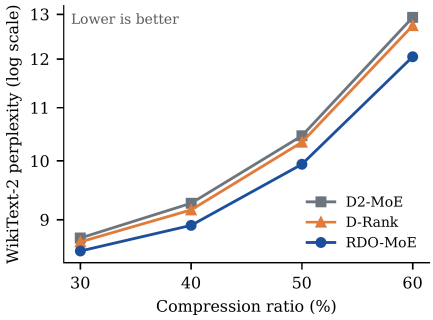


Figure 4: Qwen WikiText-2 perplexity.

Search cost. Table 3 reports the bounded 40% Qwen instance whose state space exact DP can enumerate, so its perplexities are not the full-model values of Table 2. The multiplier path reaches 8.87 perplexity against 8.82 for the exact allocation, a relative perplexity-space gap of $(8.87 - 8.82)/8.82 = 0.0057$, while using 8.12 seconds against 159.4, a 19.6× ratio. EvoESAP, the one non-exact comparator scored on the same instance, reads 8.93 and sits at 0.0125 on that scale, so the multiplier path gives up 45% of what that search gives up while running $4.85\times$ faster than it. The time gap follows from sweeping only the supported points of the frontier: exact DP enumerates partial allocations whose count can grow multiplicatively in depth, whereas each multiplier query scores every layer once.

Table 3: Solver comparison on the bounded 40% Qwen instance that exact DP can enumerate.

Method	PPL	Time (s)	Relative speed
Exact DP	8.82	159.4	1.0×
EvoESAP	8.93	39.4	4.0×
RDO-MoE	8.87	8.12	19.6×

Signed capability. Table 4 reports the signed per-task differences behind the Qwen entries. At 40% RDO-MoE is favorable on nine of eleven tasks, while WinoGrande decreases by 0.9 points and BoolQ by 1.3; at 60% those decreases grow to 1.5 and 2.1 points with nine differences still favorable. A better likelihood and a better aggregate therefore coexist with two per-task regressions, and the diagnostic reports both. The two regressions also move faster than the gains: WinoGrande and BoolQ deepen by 0.6 and 0.8 points between the two budgets while the largest favorable difference moves by 0.3, so tighter compression widens the spread across tasks rather than shifting it. At 40% the 1.3-point BoolQ loss exceeds every gain among the nine aggregated tasks, the largest being 1.1 on HellaSwag, while the aggregate lead is 0.3. More details are provided in Appendix A.15.

Table 4: Per-task Kendall τ between reconstruction and task rankings, and signed differences against D-Rank at two Qwen budgets, ordered by decreasing τ; positive favors RDO-MoE.

	ARC-E	Hella	ARC-C	PIQA	HE	OBQA	MMLU	Trivia	NQ	Wino	BoolQ
Kendall τ	.81	.79	.78	.76	.71	.62	.58	.51	.47	.20	.00
δ_t at 40%	1.0	1.1	.8	.6	1.2	.4	.5	.5	.2	-.9	-1.3
δ_t at 60%	1.3	1.4	.9	.8	1.5	.5	.7	.4	.3	-1.5	-2.1

Table 5: Factorial component response at 40% Qwen compression over surface, weighting, and order. Lower PPL and higher Score are better.

Configuration	PPL	Score
Joint/Routing/Fixed	8.91	55.71
Joint/Routing/Reversed	8.97	55.64
Joint/Uniform/Fixed	9.13	55.39
Joint/Uniform/Reversed	9.18	55.30
Indep./Routing/Fixed	9.21	55.28
Indep./Routing/Reversed	9.24	55.23
Indep./Uniform/Fixed	9.37	55.02
Indep./Uniform/Reversed	9.43	54.91

Table 6: Grid and tolerance sensitivity at 40% Qwen compression.

Condition	PPL	Score	Time (s)
Sparsity grid 3	9.04	55.3	9.38
Sparsity grid 7	8.91	55.7	9.65
Sparsity grid 15	8.89	55.8	10.21
Rank grid 4	9.07	55.2	9.32
Rank grid 8	8.91	55.7	9.65
Rank grid 16	8.90	55.7	10.34
Tolerance 10^{-2}	8.93	55.6	7.41
Tolerance 10^{-3}	8.91	55.7	9.65
Tolerance 10^{-4}	8.91	55.7	11.93

5.3 Ablation Study and Analysis

Joint allocation. Table 5 crosses three binary choices: joint versus independent surfaces, routing-weighted versus uniform expert weights, and fixed versus reversed materialization. An independent surface replaces the jointly scored distortion with the sum of the two one-at-a-time effects; the two arms differ only in the base–residual interaction. The joint/routing/fixed configuration reaches 8.91 perplexity and 55.71 aggregate score, and separating the base and residual decisions moves it to 9.21 and 55.28. Coupling therefore carries the largest share of the effect. The factorial is close to additive: the largest two-factor interaction is 0.040 perplexity against a 0.265 main effect for coupling.

Simple allocators. Table 7 replaces the multiplier with a greedy allocator that repeatedly takes the step with the largest distortion drop per byte on the same joint surfaces. It trails RDO-MoE by 0.072 perplexity at 30% and 0.318 at 60% (0.85% to 2.64%), a deficit that grows more than fourfold with compression. That deficit stays at 36 to 39% of the stage-wise D2-MoE one at every ratio: a greedy pass on the joint surfaces closes 61 to 64% of the gap to the staged pipeline, and the multiplier closes the rest. More details are provided in Appendix A.15.

Table 7: Perplexity excess over RDO-MoE at matched Qwen budgets.

Allocator	30%	40%	50%	60%
Greedy	.072	.137	.203	.318
D2-MoE	.20	.36	.52	.88

Allocator cost. After surface construction, one multiplier query costs $\mathcal{O}(\sum_l |\mathcal{C}_l|)$, or less with sorted breakpoints, which is $\mathcal{O}(L\,|\mathcal{C}|)$ under the shared per-layer grid, and the bisection adds a step count bounded by the tolerance once the bracket is expanded. Table 8 is the measured counterpart: for Qwen, construction is 155 seconds against an allocation of 9.65 seconds at 40% and 157 against 10.42 at 60%, and Mixtral uses 290 and 13.02 seconds at 40%. Each query is linear in depth at a fixed grid, and $9.65 \times (32/24) = 12.87$ against the 13.02 reported is a 1.2% excess over the depth-scaled prediction. Construction is the cold-start cost paid once per run, which is why a tighter budget moves allocation time and not build time. More details are provided in Appendix A.11.

Table 8: Resource accounting; ratios are to the uncompressed model at the two Qwen points.

Model	Compression	Build (s)	Allocate (s)	Memory	Throughput
Qwen	40%	155	9.65	.62	1.18
Qwen	60%	157	10.42	.43	1.27
Mixtral	40%	290	13.02	–	–

Routing weights and materialization order. Table 5 prices the two design choices the surface rests on. Replacing routing-weighted expert emphasis with uniform weights gives 9.13 and 55.39, while reversing materialization, which truncates each residual against the unpruned base before pruning that base, gives 8.97 and 55.64; combining all three alternatives reaches 9.43 and 54.91.

Uniform weighting therefore costs more than reversed materialization, and both cost less than decoupling, so the surface keeps the weighting it measures and the order it materializes in.

Grid and tolerance. Table 6 sweeps the two grids and the solver tolerance. Moving from the default 7 to 15 sparsity choices changes perplexity from 8.91 to 8.89 while allocation time rises from 9.65 to 10.21 seconds; tightening the tolerance from 10^{-3} to 10^{-4} leaves perplexity and score unchanged at the displayed precision while time rises to 11.93 seconds, and loosening to 10^{-2} cuts time to 7.41 seconds at a 0.02 perplexity cost. Against the ± 0.014 half-width of five calibration resamples, the coarse ends of the two grids sit 9.3 and 11.4 half-widths from the default while every refinement past it moves perplexity by at most 1.4 half-widths. The mechanism is diminishing returns near the frontier: once the grid resolves the supported points the allocator selects, extra candidates and tighter stopping add search cost without moving perplexity past 0.02. All other runs use the default grids and tolerance.

Proxy-task association. The Kendall τ row of Table 4 compares the reconstruction-based configuration order with the taskwise order over the 56 enumerated joint configurations. Association is high for ARC-Easy ($\tau = .81$), HellaSwag (.79), ARC-Challenge (.78), PIQA (.76), and HumanEval (.71); moderate for OpenBookQA (.62), MMLU (.58), TriviaQA (.51), and Natural Questions (.47); and weak for WinoGrande (.20) and BoolQ (.00), consistent with reconstruction preserving likelihood-aligned surface features but not the structure some answers depend on. On Qwen, the two regressions the diagnostic reports fall on these two weakest-association tasks, with BoolQ at $\tau = .00$ the deeper of the two. More details are provided in Appendix A.15.4.

6 Related Work

Sparse expert models. Conditional computation evolved from sparsely gated expert layers (Shazeer et al., 2017) through distributed sharding and simplified routing (Lepikhin et al., 2021; Fedus et al., 2022), and GLaM and DeepSpeed-MoE carried that design into training and inference at scale (Du et al., 2022; Rajbhandari et al., 2022). Mixtral and DeepSeekMoE brought sparse expert architectures into strong open language models (Jiang et al., 2024; Dai et al., 2024). RDO-MoE asks how their expert storage should be allocated once the sparse model and its routing are fixed.

MoE and structured compression. These methods exploit routing frequency or redundancy. D2-MoE encodes expert deltas against shared structure (Gu et al., 2025), REAP and EvoESAP prune experts (Lasby et al., 2025; Liu et al., 2026), and RFID-MoE allocates rank by routing frequency (Mi et al., 2026). MoE-I^2 and MoE-SVD add low-rank decomposition (Yang et al., 2024; Li et al., 2025); MC-MoE mixes precision (Huang et al., 2025). RDO-MoE treats the base and every expert rank as one configuration under one serialized-byte budget.

Low-rank compression. Activation-aware and truncation-aware SVD improve on unweighted matrix truncation (Yuan et al., 2023; Wang et al., 2025c;b;a; Sakr & Khailany, 2024); SAES-SVD addresses error accumulation and D-Rank varies rank across layers (Hu et al., 2026; Mi et al., 2025). These allocate rank against a reconstruction objective that returns one scalar score, which cannot separate a favorable summary from a per-task regression; evaluation practice treats the reporting protocol as part of the comparison (Biderman et al., 2024). RDO-MoE keeps the aggregate separate from taskwise signs, reporting the signed difference of every task.

7 Conclusion

RDO-MoE treats shared-base sparsity and expert-residual rank as one budgeted allocation on a single rate–distortion surface, priced by one multiplier against a global serialized-byte budget. On Qwen at 40% compression it reaches 8.91 perplexity and a 55.7 aggregate against 9.16 and 55.4 for D-Rank, and the lowest perplexity at every Mixtral budget. On the instance exact search can enumerate it stays within 0.0057 in relative perplexity at 19.6$\times$ the solver speed. The capability diagnostic keeps two regressions visible instead of one score. Indexing the base and the rank on one surface brings the jointly favorable configurations within reach of a global byte budget.

REFERENCES

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2024. URL https://arxiv.org/abs/2405.14782.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Damai Dai, Chengqi Deng, Chenggang Zhao, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of ACL*, pp. 1280–1297, 2024.

Nan Du, Yanping Huang, Andrew M. Dai, et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569, 2022.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Hugh Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337, 2023.

Hao Gu, Wei Li, Lujun Li, Qiyuan Zhu, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for MoE-based LLMs compression. In *International Conference on Machine Learning*, 2025.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Xing Hu, Dawei Yang, Yuan Cheng, Zhixuan Chen, and Zukang Xu. SAES-SVD: Self-adaptive suppression of accumulated and local errors for SVD-based LLM compression. *arXiv preprint arXiv:2602.03051*, 2026.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts LLMs gains more. In *International Conference on Learning Representations*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression, 2025. URL https://arxiv.org/abs/2510.13999.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Lujun Li, Qiyuan Zhu, Jiacheng Wang, Xiaoyu Qin, Wei Li, Hao Gu, Sirui Han, and Yike Guo. SubMoE: Efficient mixture-of-expert LLMs compression via subspace expert merging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i27.39464.

Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. MoE-SVD: Structured mixture-of-experts LLMs compression via singular value decomposition. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ho7ZUS1z8A.

Zongfang Liu, Shengkun Tang, Boyang Sun, Zhiqiang Shen, and Xin Yuan. Evoesap: Non-uniform expert pruning for sparse moe, 2026. URL https://arxiv.org/abs/2603.06003.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Zhendong Mi, Bian Sun, Grace Li Zhang, and Shaoyi Huang. Layer-wise dynamic rank for compressing large language models. *arXiv preprint arXiv:2509.25622*, 2025.

Zhendong Mi, Yixiao Chen, Pu Zhao, Xiaodong Yu, Hao Wang, Yanzhi Wang, and Shaoyi Huang. Effective MoE-based LLM compression by exploiting heterogeneous inter-group experts routing frequency and information density. *arXiv preprint arXiv:2602.09316*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of EMNLP*, pp. 2381–2391, 2018.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 01 1960. ISSN 0033-5606. doi: 10.1093/qmath/11.1.50. URL https://doi.org/10.1093/qmath/11.1.50.

A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, 1998. doi: 10.1109/79.733495.

Qwen Team. Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters. qwenlm.github.io, 2024. URL https://qwenlm.github.io/blog/qwen-moe.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *International Conference on Machine Learning*, pp. 18332–18346, 2022.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020.

Charbel Sakr and Brucek Khailany. ESPACE: Dimensionality reduction of activations for model compression. In *Advances in Neural Information Processing Systems 37*, 2024. doi: 10.52202/079017-0556.

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

Yair Shoham and Allen Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, 1988.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024.

Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=kws76i5XB8.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. SVD-LLM V2: Optimizing singular value truncation for large language model compression. In *Proceedings of NAACL-HLT*, pp. 4287–4296, 2025b.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=LNYIUouhdt.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. MoE-I^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of EMNLP*, pp. 10456–10466, 2024.

Lu Yin, You Wu, Zhenyu Zhang, et al. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *International Conference on Machine Learning*, pp. 57101–57115, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models, 2023. URL https://arxiv.org/abs/2312.05821.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of ACL*, pp. 4791–4800, 2019.

AI USE STATEMENT

We used generative AI tools to translate the text (Claude Opus 5.0), to check grammar and spelling and polish the wording (GPT-5.6), to guide the drawing of the figures (Gemini), and to check the content (Grok). We did not use generative AI tools to modify code. We have reviewed all AI-assisted work: we checked every reference for hallucinated citations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The details needed to reproduce the results of this paper, including the allocation procedure, the serialized-byte rate model, the hyperparameters, the evaluation metrics and the software setup, are disclosed in this appendix.

APPENDIX

Table S1 lists the contents of each part of the appendix.

Table S1: What each part of the appendix holds.

Section	Contents
Appendix A.0	Limitations
Appendix A.1	Extended related work
Appendix A.2	Motivation panels in detail
Appendix A.3	Formal statements and proofs
Appendix A.4	The allocation procedure in pseudocode
Appendix A.5	System and software environment
Appendix A.6	Datasets and evaluation suite
Appendix A.7	Serialized-byte rate model
Appendix A.8	Baseline configurations
Appendix A.9	Evaluation metrics
Appendix A.10	Reproduction details
Appendix A.11	Complexity and implementation notes
Appendix A.12	When joint allocation pays off
Appendix A.13	Additional experimental detail
Appendix A.14	Broader impacts and responsible use
Appendix A.15	Extended experimental results
Appendix A.16	Interaction contrasts of the factorial design

APPENDIX A.0: LIMITATIONS

Evaluation corpora. The paper lacks perplexity on held-out corpora: calibration, routing statistics and perplexity all come from WikiText-2. A compressed model meets text other than its calibration data, such as web text, encyclopedic text or code, and perplexity on such corpora is what shows whether the method ordering and the margin of the joint allocator hold away from the calibration domain.

Allocator scale. The paper lacks a controlled measurement of how surface construction time and memory grow with the number of experts per layer; the two backbones give only two points on that axis, and they also differ in layer count and matrix size. A sweep over the number of experts with everything else held fixed is what isolates the growth rate of the construction step.

Mild compression. The paper lacks a comparison of joint and sequential allocation at compression ratios below 30%. Light compression is where the base is pruned least, and comparing the two allocations there shows whether the joint surface still changes the outcome when the budget is loose.

APPENDIX A.1: EXTENDED RELATED WORK

This section expands the positioning of Sec. 6 across the method families the comparison draws on.

Sparse expert models. Conditional computation developed from sparsely gated expert layers (Shazeer et al., 2017) through the sharded routing of GShard and Switch Transformers (Lepikhin et al., 2021; Fedus et al., 2022) and the scaled training and serving of GLaM and DeepSpeed-MoE (Du et al., 2022; Rajbhandari et al., 2022), with Mixtral, DeepSeekMoE, and Qwen1.5-MoE bringing sparse experts into strong open models (Jiang et al., 2024; Dai et al., 2024; Qwen Team,

2024). Reproducible evaluation practice for these models (Biderman et al., 2024) fixes the measurement surface this work inherits. RDO-MoE takes the sparse architecture and evaluation protocol as fixed and asks only how expert storage should be allocated afterwards, so this line supplies the backbone context.

MoE and structured compression. D2-MoE encodes expert deltas relative to a shared base (Gu et al., 2025), and MoE-I^2 combines inter-expert pruning with intra-expert low-rank decomposition (Yang et al., 2024), both being the closest same-family references; REAP and EvoESAP decide whole-expert or behavior-informed pruning (Lasby et al., 2025; Liu et al., 2026), and RFID-MoE allocates per-expert rank from routing frequency and effective rank (Mi et al., 2026); MC-MoE and MoE-SVD expose heterogeneous per-expert compression opportunities (Huang et al., 2025; Li et al., 2025); Sub-MoE merges experts through a shared subspace (Li et al., 2026); and SparseGPT, Wanda, and OWL are adjacent one-shot dense-model pruning methods (Frantar & Alistarh, 2023; Sun et al., 2024; Yin et al., 2024). RDO-MoE differs by treating the shared-base sparsity and every expert-residual rank as one finite per-layer configuration under a single global serialized-byte budget, rather than pruning experts or fixing a base before truncating residuals.

Low-rank compression and task coverage. Activation-aware and truncation-aware SVD (Yuan et al., 2023; Wang et al., 2025c;b) improve on unweighted truncation (Eckart & Young, 1936; Mirsky, 1960); SAES-SVD targets error accumulation and D-Rank varies rank across layers (Hu et al., 2026; Mi et al., 2025), both of which appear as baselines. The task suite complements likelihood with MMLU, BoolQ, HellaSwag, PIQA, WinoGrande, ARC, OpenBookQA, Natural Questions, TriviaQA, and HumanEval (Hendrycks et al., 2021; Clark et al., 2019; Zellers et al., 2019; Bisk et al., 2020; Sakaguchi et al., 2020; Clark et al., 2018; Mihaylov et al., 2018; Kwiatkowski et al., 2019; Joshi et al., 2017; Chen et al., 2021).

Rate-distortion allocation lineage. The allocation mechanism draws on classical rate–distortion optimization, where a single Lagrange multiplier trades distortion against rate and decouples an allocation across coding units by selecting each unit independently at a shared multiplier (Everett, 1963; Shoham & Gersho, 1988; Ortega & Ramchandran, 1998). RDO-MoE applies that mechanism to per-layer tables of jointly scored shared-base sparsity and residual-rank pairs, priced in serialized bytes.

Appendix A.2: Motivation Panels in Detail

The two panels of Figure 2 are reproduced here one per figure, together with the quantities plotted and the reading each panel supports.

Appendix A.2.1: Left panel, the cost of staging

Figure S1 plots perplexity at compression ratios of 30, 40, 50 and 60 percent; the vertical axis is WikiText-2 perplexity on a logarithmic scale. Two published orders are plotted at every ratio, a stage-wise delta decomposition that fixes the shared base before the rank and a rank-only allocation that fixes the rank instead. The bars on the right axis give the relative distance between them, each taken against the larger of the two perplexities at that ratio, 0.69, 1.19, 1.15 and 1.47 percent across the four ratios. The reading the panel supports is that the two orders stand furthest apart at the tightest budget. The panel plots published methods only.

Appendix A.2.2: The solver table of the motivation

Table 1 of the motivation is the same comparison as Table S8, with the row of the proposed method withheld. Both report the bounded 40% Qwen instance whose state space the exact route can enumerate, and both give WikiText-2 perplexity and allocation wall time on that restricted instance rather than on the full model. The relative speed column is taken against the exact search. The exact route is admitted on this instance because its nondominated partial states stay within the admissibility bound of Table S6.

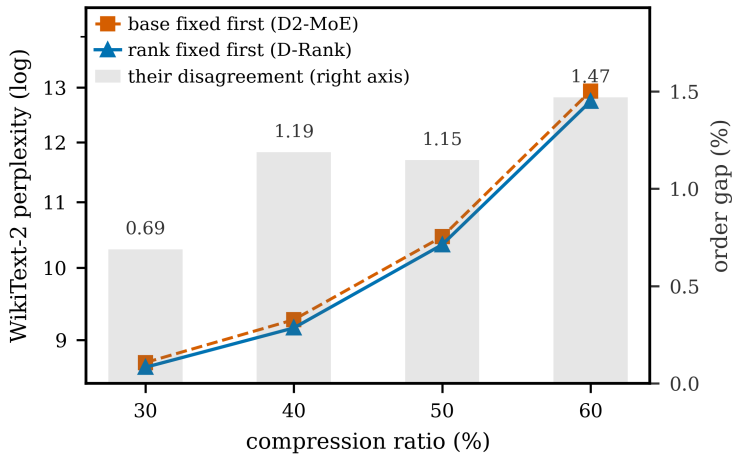


Figure S1: Perplexity against compression ratio for two published staged orders on Qwen, with their relative distance as bars on the right axis. The stage-wise curve is D2-MoE and the rank-only curve is D-Rank.

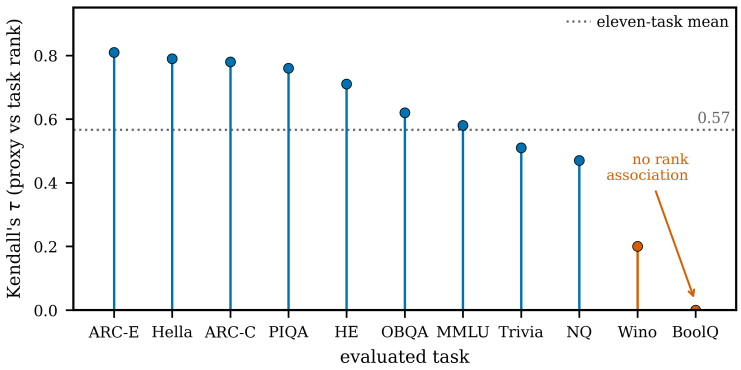


Figure S2: Rank agreement between the reconstruction proxy and task performance, one stem per evaluated task, ordered by agreement. The dotted line is the eleven-task mean.

Appendix A.2.3: Right panel, the proxy and the tasks

Figure S2 plots Kendall's τ between the configuration ranking induced by reconstruction error and the ranking induced by task performance, one stem per evaluated task, ordered by that agreement. The dotted horizontal line marks the eleven-task mean of 0.57. Agreement ranges from 0.81 on ARC-Easy down to 0.20 on WinoGrande and 0.00 on BoolQ, so the proxy is not uniformly informative about task rank and on the weakest task shows no rank association. The reliability of the reconstruction proxy therefore varies by task, and the spread is wide enough for an aggregate over tasks to be silent about where the agreement failed.

Appendix A.3: Formal Statements and Proofs

The formal statements referenced by the problem formulation (Sec. 3) and the method (Sec. 4) establish that the joint surface and the multiplier path are well posed.

Singular-tail derivation. For fixed s and r, the best rank-r residual error equals the discarded singular tail $\sum_{j>r} \sigma_j(\Delta_{li}(s))^2$, and that residual matrix depends on s through Eq. 2. When the tail energy moves with s, the rank-distortion term is not invariant to the shared-base decision. The proposition below states that dependence formally.

Proposition S1 (Conditional spectral dependence). *For fixed s and r, the minimum squared residual error is $\sum_{j>r} \sigma_j(\Delta_{li}(s))^2$. If this tail energy changes with s for any expert and rank, the rank-distortion term is not invariant to the shared-base decision.*

Proof. By the Eckart-Young-Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960), the minimum of $\|\Delta - T\|_F^2$ over matrices T of rank at most r equals $\sum_{j>r} \sigma_j(\Delta)^2$. Applying this with $\Delta = \Delta_{li}(s)$ gives the first statement. By Eq. 2 the matrix $\Delta_{li}(s) = W_{li} - P_s(W_{lb})$, and hence each singular value $\sigma_j(\Delta_{li}(s))$, is a function of s, so the minimum error is the map $s \mapsto \sum_{j>r} \sigma_j(\Delta_{li}(s))^2$. If two admissible values $s_a \neq s_b$ give different tail sums for some expert i and rank r, that map is nonconstant, so the rank-distortion term is not invariant to the shared-base decision. □

Mixed-difference diagnostic. Non-separability requires more than dependence. Under the shared-per-layer-rank enumeration of Sec. 4, write $D_l(s, r)$ for the layer distortion at base sparsity s and common residual rank r. On a finite grid, an additive form $D_l(s, r) = f_l(s) + g_l(r)$ is rejected by a nonzero mixed difference

$$\Gamma_l = D_l(s_a, r_a) + D_l(s_b, r_b) - D_l(s_a, r_b) - D_l(s_b, r_a). \tag{S1}$$

Substituting Eq. 3 into Eq. S1 and taking $r_a < r_b$ without loss of generality, each inner difference telescopes to one band of the spectrum, $D_l(s, r_a) - D_l(s, r_b) = \sum_i \pi_{li} \sum_{j=r_a+1}^{r_b} \sigma_j(\Delta_{li}(s))^2$, the routing-weighted energy the base s leaves between the two ranks, giving

$$\Gamma_l = \sum_i \pi_{li} \sum_{j=r_a+1}^{r_b} \Big[\sigma_j(\Delta_{li}(s_a))^2 - \sigma_j(\Delta_{li}(s_b))^2\Big]. \tag{S2}$$

The interaction is therefore the difference between what the two bases leave in one band of the residual spectrum, and it vanishes only where the two leave the same routing-weighted energy there. This diagnostic isolates the joint-surface claim from all downstream task metrics.

Proposition S2 (Monotone selected rate). *If $0 \leq \lambda_1 < \lambda_2$, then $\sum_l R_l(c_l(\lambda_2)) \leq \sum_l R_l(c_l(\lambda_1))$.*

Proof. Fix a layer l and write $c_1 = c_l(\lambda_1)$ and $c_2 = c_l(\lambda_2)$, dropping the layer index on D_l and R_l. Optimality of c_1 at λ_1 gives $D(c_1) + \lambda_1 R(c_1) \leq D(c_2) + \lambda_1 R(c_2)$, and optimality of c_2 at λ_2 gives $D(c_2) + \lambda_2 R(c_2) \leq D(c_1) + \lambda_2 R(c_1)$. Adding the two inequalities cancels the distortion terms and leaves $(\lambda_2 - \lambda_1)\big(R(c_2) - R(c_1)\big) \leq 0$. Since $\lambda_2 > \lambda_1$, this forces $R(c_2) \leq R(c_1)$. The argument applies to every layer independently, and summing the L per-layer inequalities gives the stated bound on the total selected rate. □

Appendix A.4: The Allocation Procedure in Pseudocode

Algorithm S1 states in one place the procedure that Sec. 4 describes in prose, and runs the three components as a single pass in four stages. Stage 1 freezes the routing frequencies before anything is selected, prices every admitted sparsity–rank pair of a layer as one rate–distortion point, and keeps only the nondominated ones, so the sparsity and the rank are priced together rather than in sequence. Stage 2 first asks whether the budget can be met at all, and whether the unpriced selection already meets it; otherwise it doubles a trial upper multiplier until the rate it selects fits, bisects that bracket to the requested relative width, and returns the lowest-distortion feasible configuration it visited, each midpoint test being decided by Proposition S2. Every member of $\mathcal{F}$ passes an explicit budget test before it is admitted, which is what makes the returned configuration feasible by construction. Stage 3 realizes that configuration in one pass and serializes it under the byte convention that built R_l, recording the unspent budget as ϵ_B. Stage 4 scores the realized model against the reference and reports the signed movement of every task, outside the allocation objective. More details about the cost of each stage are provided in Appendix A.11.

Appendix A.5: System and Software Environment

Every construction, allocation, and evaluation run reported in this paper was executed on a single NVIDIA A100 80 GB GPU under CUDA 12.8 with PyTorch 2.11. Model weights and the forward pass are bfloat16 throughout, while the residual factorization is computed in float32 and the singular

Algorithm S1: RDO-MoE in one pass: joint per-layer surfaces, multiplier-indexed allocation, and capability diagnostic

Require: Expert matrices W_{li} and fixed shared bases W_{lb}; deterministic base operator P_s; finite nonempty grids $\mathcal{S}$ and $\mathcal{R}_0$, so that $\mathcal{C}_l = \mathcal{S} \times \mathcal{R}_0$; calibration subset
Require: Global serialized-byte budget B under one fixed byte convention; trial upper multiplier $\lambda_0 > 0$; relative tolerance $\eta \in (0, 1)$
Require: Reference c_{ref} at the same budget, evaluated under the same template, split, and aggregation rule; tasks $\mathcal{T}$ with metrics M_t and directions $q_t \in \{+1, -1\}$; nonempty commensurable family $\mathcal{G} \subseteq \mathcal{T}$
Ensure: Serialized checkpoint at $\hat{c}$, budget slack ϵ_B, and signed readout $(\{\delta_t\}, \bar{\delta}, \mathcal{T}^-)$; or INFEASIBLE

Stage 1: Joint per-layer surfaces

Measure the routing frequencies $\pi_{li} \geq 0$, $\sum_i \pi_{li} = 1$, on the calibration subset and freeze them
for each layer l **do**
  for each $c_l = (s, r) \in \mathcal{C}_l$ **do**
    $\Delta_{li}(s) \leftarrow W_{li} - P_s(W_{lb})$ for $i = 1, \ldots, E_l$ ▷ Eq. 2
    $R_l(c_l) \leftarrow$ serialized bytes of $P_s(W_{lb})$, of the rank-r factors of $T_r(\Delta_{li}(s))$, $i = 1, \ldots, E_l$, and of the per-layer header
    $D_l(c_l) \leftarrow \sum_{i=1}^{E_l} \pi_{li} \left\| \Delta_{li}(s) - T_r(\Delta_{li}(s)) \right\|_F^2$ ▷ Eq. 3 at $r_{li} = r$
  $\mathcal{P}_l \leftarrow$ nondominated configurations of $\mathcal{C}_l$, keeping their enumeration indices ▷ Eq. 4

Stage 2: Multiplier-indexed allocation

Throughout, $c_l(\lambda)$ is the minimizer of Eq. 5 on $\mathcal{P}_l$, ties broken by lower R_l and then by enumeration index, and $c(\lambda) = (c_l(\lambda))_l$
if $B < \sum_l \min_{c \in \mathcal{P}_l} R_l(c)$ **then**
  return INFEASIBLE ▷ no assignment fits the budget
if $\sum_l R_l(c_l(0)) \leq B$ **then**
  $\mathcal{F} \leftarrow \{c(0)\}$ ▷ the unpriced selection already fits
else
  $(\lambda_{\mathrm{lo}}, \lambda_{\mathrm{hi}}) \leftarrow (0, \lambda_0)$
  while $\sum_l R_l(c_l(\lambda_{\mathrm{hi}})) > B$ **do**
    $(\lambda_{\mathrm{lo}}, \lambda_{\mathrm{hi}}) \leftarrow (\lambda_{\mathrm{hi}}, 2\lambda_{\mathrm{hi}})$ ▷ the grids are finite, so a large enough multiplier selects the minimum rate admitted above
  $\mathcal{F} \leftarrow \{c(\lambda_{\mathrm{hi}})\}$ ▷ feasible bracketing endpoint, counted as visited
  while $(\lambda_{\mathrm{hi}} - \lambda_{\mathrm{lo}})/\lambda_{\mathrm{hi}} \geq \eta$ **do**
    $\lambda \leftarrow (\lambda_{\mathrm{lo}} + \lambda_{\mathrm{hi}})/2$
    if $\sum_l R_l(c_l(\lambda)) \leq B$ **then**
      $\lambda_{\mathrm{hi}} \leftarrow \lambda$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{c(\lambda)\}$ ▷ by Proposition S2
    else
      $\lambda_{\mathrm{lo}} \leftarrow \lambda$
$\hat{c} \leftarrow \arg\min_{c \in \mathcal{F}} \sum_l D_l(c_l)$ ▷ Eq. 7

Stage 3: One-pass realization and serialization

for each layer l, with $\hat{c}_l = (s_l, r_l)$ **do**
  Recompute $P_{s_l}(W_{lb})$ and $\Delta_{li}(s_l)$, and serialize the base with the factors of $T_{r_l}(\Delta_{li}(s_l))$, $i = 1, \ldots, E_l$, under the byte convention of R_l
$\epsilon_B \leftarrow |B - \sum_l R_l(\hat{c}_l)| / \max(B, 1)$ ▷ Eq. 8

Stage 4: Capability diagnostic, outside the objective

$\delta_t \leftarrow q_t \left[M_t(\hat{c}) - M_t(c_{\mathrm{ref}}) \right]$ for $t \in \mathcal{T}$, with $M_t(\hat{c})$ read on the serialized model ▷ Eq. 9
$\bar{\delta} \leftarrow |\mathcal{G}|^{-1} \sum_{t \in \mathcal{G}} \delta_t$; $\mathcal{T}^- \leftarrow \{t \in \mathcal{T} : \delta_t < 0\}$ ▷ Eq. 10
return the serialized checkpoint, ϵ_B, and $(\{\delta_t\}, \bar{\delta}, \mathcal{T}^-)$

values are folded into one factor, so the spectrum used to build D_l does not depend on the storage precision of the compressed model. Table S2 lists the stack.

Model assets are pinned to fixed releases so that a reproducer resolves the same weights. For Qwen1.5-MoE-A2.7B (Qwen Team, 2024) we use the release at short commit `ce35c8d`, distributed under its custom Tongyi-Qianwen license; for Mixtral-8x7B-v0.1 (Jiang et al., 2024) we use the release at short commit `5802963`, distributed under Apache-2.0. Capability scores use the lm-evaluation-harness v0.4.3 lineage (Biderman et al., 2024) with `use_fast=false` tokenization and the pad token set to the end-of-sequence token when the tokenizer defines none.

Table S2: Execution environment for all reported runs.

Component	Setting
GPU	NVIDIA A100 80 GB
CUDA	12.8
Framework	PyTorch 2.11
Model precision	bfloat16
Factorization precision	float32

APPENDIX A.6: DATASETS AND EVALUATION SUITE

Aggregate definition. Restated from Sec. 5. The eleven evaluated tasks fall into two families whose metrics are not commensurable: nine are scored as accuracy, while Natural Questions and HumanEval are open-ended generation metrics whose observed values sit far lower. The aggregate reported throughout is therefore the unweighted mean of the nine accuracy-scored tasks, with the two generation tasks reported separately in Table S9 and signed differences reported for all eleven in Table S11. Calibration and expert-frequency statistics use the WikiText-2 training subset, consistent with the delta-decomposition preprocessing lineage (Gu et al., 2025).

Table S3: Evaluation suite: nine accuracy-scored tasks define the aggregate; Natural Questions and HumanEval are reported separately.

Dataset	Role	Config	Splits
WikiText-2	Perplexity, calibration	wikitext-2-raw-v1	train/val/test
ARC-Easy	Zero-shot task	ARC-Easy	train/val/test
ARC-Challenge	Zero-shot task	ARC-Challenge	train/val/test
HellaSwag	Zero-shot task	default	train/val/test
PIQA	Zero-shot task	plain_text	train/val/test
WinoGrande	Zero-shot task	winogrande_xl	train/val/test
BoolQ	Zero-shot task	default	train/val
OpenBookQA	Zero-shot task	main	train/val/test
MMLU	Zero-shot task	all	aux/dev/val/test
TriviaQA	Zero-shot task	rc.nocontext	train/val/test
NaturalQuestions	Zero-shot task	default	train/val
HumanEval	Zero-shot task	as published	as published

Table S3 documents the corpora behind the comparisons. WikiText-2 supplies both the language-model perplexity readout and the calibration and expert-frequency statistics, consistent with the delta-decomposition preprocessing lineage. The remaining suites supply the taskwise capability readings.

APPENDIX A.7: SERIALIZED-BYTE RATE MODEL

All methods at an operating point are compared under identical accounting, so this section states the components of the serialized-byte rate $R_l(c_l)$ and how they are counted.

APPENDIX A.7.1: BASE OPERATOR AND RESIDUAL TRUNCATION

In the reported construction, each admitted s satisfies $k_s = \lfloor s\, d_{\text{out}} d_{\text{in}} \rfloor$ for a weight matrix of shape $d_{\text{out}} \times d_{\text{in}}$, and P_s zeros exactly k_s entries of W_{lb} in increasing absolute-magnitude order, breaking threshold ties by row-major position. The residual truncation uses a float32 singular-value decomposition and stores two rank-r factors with the singular values absorbed into one factor. These three choices are what make the configuration grid deterministic: for a given (s, r) every method sees the same zeroed entries, the same factorization, and the same base.

Table S4: Restated from the main paper. Compression quality across backbones and methods; lower PPL and higher aggregate score are better. Bold marks the best value in each column within a backbone block.

		30%		40%		50%		60%	
Model	Method	PPL	Score	PPL	Score	PPL	Score	PPL	Score
Qwen	Uniform+Magnitude	9.71	54.7	10.84	52.9	13.06	50.1	17.84	46.2
Qwen	SAES-SVD	8.95	56.0	9.62	54.6	11.18	52.1	14.35	48.4
Qwen	EvoESAP	8.78	56.5	9.31	55.1	10.62	52.7	13.26	49.0
Qwen	D-Rank	8.65	56.8	9.16	55.4	10.34	53.0	12.74	49.4
Qwen	D2-MoE	8.71	56.7	9.27	55.2	10.46	52.8	12.93	49.2
Qwen	**RDO-MoE**	**8.51**	**57.0**	**8.91**	**55.7**	**9.94**	**53.2**	**12.05**	**49.7**
Mixtral	EvoESAP	5.76	64.9	5.96	63.3	6.47	60.6	7.40	57.4
Mixtral	D-Rank	5.68	64.9	5.88	63.6	6.33	**60.9**	7.18	57.7
Mixtral	D2-MoE	6.00	64.8	6.22	63.3	6.70	60.4	7.62	57.4
Mixtral	**RDO-MoE**	**5.33**	**65.2**	**5.49**	**64.1**	**5.86**	60.8	**6.58**	**58.3**

A layer configuration $c_l = (s_l, r_{l1}, \ldots, r_{lE_l})$ serializes into three parts. First, the shared base $P_{s_l}(W_{lb})$ retained after magnitude pruning contributes its surviving nonzero weights together with the index structure a sparse format must store; a configuration that zeros k_{s_l} entries retains $d_{\text{out}}d_{\text{in}} - k_{s_l}$ base values plus that format's index overhead. Second, each expert residual is stored as its two rank-r_{li} factors, with the singular values folded into one factor, contributing on the order of $r_{li}(d_{\text{out}} + d_{\text{in}})$ stored numbers per expert. Third, a fixed per-layer header records the chosen sparsity level and the per-expert ranks so that the configuration is self-describing at load time. The same byte-rounding rule is applied to all three parts for every candidate, so a change of representation or rounding defines a different rate table rather than a different point on the same table.

Two accounting choices matter for comparability. The sparse-index overhead is charged to the base part rather than assumed free, because a raw parameter count does not show it at all; charging it per entry puts every method on one accounting for the same retained count. Under the rate card of Table S6 the charge is 1 bit of index per entry against 16 bits of value per retained entry, charged on the same terms for every method. The residual factor count uses the shared-per-layer rank of the reported enumeration, so the residual contribution reduces to $E_l\, r_l(d_{\text{out}} + d_{\text{in}})$ at the reported grid. With $n = d_{\text{out}}d_{\text{in}}$, the rate of a configuration (s_l, r_l) is

$$R_l(s_l, r_l) = \left\lceil \tfrac{1}{8}\left[16\,(n - k_{s_l}) + n + 16\,E_l\, r_l(d_{\text{out}} + d_{\text{in}}) + 8 \cdot 64\right] \right\rceil \text{ bytes},$$

the four terms being the retained base values, the one-bit index per entry, the residual factors, and the 64-byte header, with r_l capped at $\min(d_{\text{out}}, d_{\text{in}})$. All methods serialize under one fixed sparse-container and header encoding with the same byte rounding, so every candidate is scored under the identical representation the body asserts.

Appendix A.8: Baseline Configurations

Uniform+Magnitude is the simple storage control, built from unstructured magnitude pruning (Han et al., 2015) at a uniform rate. SAES-SVD (Hu et al., 2026) and D-Rank (Mi et al., 2025) represent low-rank allocation, D2-MoE (Gu et al., 2025) represents delta decomposition, and EvoESAP (Liu et al., 2026) represents nonuniform expert pruning. A greedy allocator provides the simple-allocator control: it starts each layer at its cheapest frontier point and repeatedly takes the step with the largest distortion drop per byte until the budget is exhausted. Exact dynamic programming is admitted only on the bounded solver instance, where its state space stays within the admissibility bound recorded with the reported hyperparameters.

The Qwen curve carries the full comparator set; the Mixtral comparison carries the three closest comparators together with this paper's allocator, with the uniform storage control and SAES-SVD reported on Qwen only. Every comparison point is produced under the protocol of this paper rather than carried over from a method's own published configuration, since parameter counts alone do not capture sparse indices, residual factors, or byte rounding.

APPENDIX A.9: EVALUATION METRICS

Language-model quality is the perplexity of the compressed model over the WikiText-2 test split under a concatenated sliding window, which for a token sequence $x_{1:N}$ scored by the model distribution p is

$$\mathrm{PPL} = \exp\Big(-\tfrac{1}{N}\sum_{n=1}^{N}\log p(x_n \mid x_{<n})\Big), \tag{S3}$$

evaluated on the split and tokenization convention introduced with that corpus (Merity et al., 2017). Lower is better.

Nine of the eleven evaluated tasks are scored as accuracy, six under the vendored harness templates (Biderman et al., 2024) and three as custom additions under the same zero-shot policy. Writing A_t for the accuracy on task t and $\mathcal{G}$ for that nine-task family, the aggregate reported throughout is the unweighted mean

$$\bar{A} = \frac{1}{|\mathcal{G}|}\sum_{t\in\mathcal{G}} A_t. \tag{S4}$$

Higher is better. The two remaining tasks stay outside Eq. S4 because their scales are not commensurable with accuracy.

Natural Questions is scored by exact match over free-form generation, the fraction of the Q evaluated items whose normalized prediction $\hat{y}_i$ equals a normalized reference y_i (Kwiatkowski et al., 2019),

$$\mathrm{EM} = \frac{1}{Q}\sum_{i=1}^{Q}\mathbf{1}\big[\hat{y}_i = y_i\big]. \tag{S5}$$

HumanEval is scored by pass@1, the expected fraction of problems solved by a single sample; with n samples drawn per problem and c of them passing the unit tests, the unbiased estimator of Chen et al. (2021) at $k = 1$ reduces to

$$\mathrm{pass@1} = \mathop{\mathbb{E}}_{\text{problems}}\Big[\frac{c}{n}\Big]. \tag{S6}$$

Both are reported separately from Eq. S4.

Signed taskwise movement is the direction-normalized paired difference δ_t of Eq. 9, reported for all eleven tasks so that a favorable aggregate cannot hide a negative task. Rank agreement between the reconstruction-based configuration order and the taskwise order is summarized by Kendall's rank correlation coefficient τ, computed over the enumerated joint configurations. Resource accounting covers curve-construction time, allocation time, memory normalized to the uncompressed model, and throughput; solver quality covers wall time, relative speed, and the relative perplexity-space gap to the exact-DP allocation.

APPENDIX A.10: REPRODUCTION DETAILS

This section complements Sec. 5 with the deterministic choices and reconstruction sequence. The finite configuration grid consists of the admitted shared-base sparsities and residual ranks; changing either grid defines a different comparison. The deterministic choices behind that grid are listed with the reported hyperparameters in Table S6; the base operator and the residual truncation are stated formally in Appendix A.7.1. Methods compared at one operating point use the same byte rounding, calibration split, task template, and aggregation rule. Every comparison point is produced under that protocol on the settings of this paper, rather than carried over from a method's own published configuration.

A reconstruction attempt should first verify identical source weights and dataset revisions, rebuild the layer surfaces, remove dominated points, trace multiplier breakpoints, serialize the chosen configuration under the same representation, and then evaluate likelihood and task scores without using those scores for allocation.

Reconstruction items. Table S5 points each reconstruction item to a paper section: configuration definitions are in the formulation, the byte-accounting convention is in Appendix A.7, comparison protocols are in Sec. 5, and deterministic choices plus a reconstruction checklist appear in this appendix. Dataset and source definitions are tied to their primary references.

Table S5: Reproducibility materials.

Item	Stated in
Configuration definitions	Problem formulation and Appendix A.10
Byte-accounting convention	Appendix A.7
Configuration comparison	Sec. 5
Deterministic-choice rationale	Appendix A.10
Reproduction checklist	Appendix A.10
Dataset and source definitions	Sec. 5 and References

Evaluation protocol specification. The capability suite is scored under the lm-evaluation-harness task templates (Biderman et al., 2024), following the vendored v0.4.3 lineage used by the delta-decomposition reference (Gu et al., 2025), with the task identifiers `openbookqa`, `arc_easy`, `winogrande`, `hellaswag`, `arc_challenge`, and `piqa`. These six zero-shot task templates are the ones covered directly by the vendored harness lineage. The further reported capabilities (BoolQ, MMLU, TriviaQA, Natural Questions, and HumanEval) are custom additions to the shared evaluation suite, and they follow the shared zero-shot policy of this section. HumanEval is the one task that carries a setting beyond that policy, since pass@1 is an estimate over samples rather than a score over a fixed answer set: it is estimated with the benchmark's own unbiased estimator (Chen et al., 2021) from $n = 10$ samples per problem over the 164 problems, at temperature 0.2 and nucleus 0.95. The few-shot policy is `num_fewshot=` 0 for every task. The tokenizer is loaded with `use_fast=false`, and when the tokenizer supplies no pad token the end-of-sequence token is used as the pad token. WikiText-2 perplexity (Merity et al., 2017) follows the concatenated-corpus sliding-window convention of the one-shot compression literature (Frantar & Alistarh, 2023; Gu et al., 2025). Calibration and the reported statistics use the WikiText-2 subset.

Version and asset pinning. To let a reproducer obtain the same model assets, we record the release revisions used. For Qwen1.5-MoE-A2.7B (Qwen Team, 2024) we use the release at short commit `ce35c8d`, distributed under its custom Tongyi-Qianwen license. For Mixtral-8x7B-v0.1 (Jiang et al., 2024) we use the release at short commit `5802963`, distributed under Apache-2.0. Capability scores use the lm-evaluation-harness v0.4.3 lineage (Biderman et al., 2024). Calibration draws 512 sequences of 2048 tokens from the WikiText-2 (raw) training split, and WikiText-2 perplexity is reported on the test split (Merity et al., 2017). All timing and memory measurements were collected on a single NVIDIA A100 80 GB GPU under CUDA 12.8 and PyTorch 2.11, with bfloat16 model weights. Surface construction takes 155 seconds for Qwen1.5-MoE-A2.7B and 290 seconds for Mixtral-8$\times$7B.

Reproduction manifest. A reproducer must supply four inputs to reconstruct the comparison: (i) the base model and tokenizer at the same revision, with the tokenizer flags above; (ii) the calibration subset, namely the WikiText-2 subset used for calibration and statistics; (iii) the task harness with its templates, namely the lm-evaluation-harness v0.4.3 lineage (Biderman et al., 2024) at zero shots, covering the six vendored task identifiers listed above together with the five further capabilities (BoolQ, MMLU, TriviaQA, Natural Questions, and HumanEval) scored under the same zero-shot policy, which is what the nine-task aggregate and the eleven signed differences require; and (iv) the allocator configuration, namely the admitted sparsity and rank grids, the fixed serialized-byte representation, and the multiplier tolerance.

Table S6 collects the allocator and evaluation hyperparameters used for the reported runs, including the solver and serialization settings that the formulation, method, and setup do not spell out individually.

Appendix A.11: Complexity and Implementation Notes

This section expands the cost statements of Sec. 5.

The allocator separates a one-time surface-construction cost from a per-query allocation cost. Surface construction evaluates every candidate in each layer once, applying the base operator, recomputing the residual, forming the requested factors, and recording the rate–distortion pair; this is the

Table S6: Allocator and evaluation hyperparameters used for the reported runs. Entries not spelled out in the formulation, method, or experimental setup are fixed here.

Hyperparameter	Value
Base-sparsity grid $\mathcal{S}$	7 levels: $0, 0.1, \ldots, 0.6$
Residual-rank grid $\mathcal{R}_0$	8 levels: $0, 4, 8, 16, 32, 64, 128, 256$
Per-layer configurations $\lvert\mathcal{C}_l\rvert = \lvert\mathcal{S}\rvert \times \lvert\mathcal{R}_0\rvert$	56
Expert-rank sharing	one common rank per layer
Base operator P_s	magnitude pruning, ties by row-major position
Residual factorization	float32 SVD, singular values folded into one factor
Tie rule in Eq. 5	lower serialized rate, then enumeration index
Solver tolerance (default)	10^{-3}
Compression ratios (Qwen)	30%, 40%, 50%, 60%
Compression ratios (Mixtral)	30%, 40%, 50%, 60%
Few-shot setting	`num_fewshot`$= 0$ (all tasks)
Tokenizer	`use_fast=false`; pad token $=$ eos when absent
Perplexity convention	concatenated sliding window, WikiText-2 test
Calibration set	512 sequences $\times$ 2048 tokens (WikiText-2)
Value bytes per retained entry	2
Index cost per entry	1.0 bit
Per-layer header	64 bytes
Byte rounding	ceiling
Exact-DP admissibility bound	$\leq$ 200,000 nondominated partial states
Upper-multiplier start	10^{-6}, growth factor 2.0
Multiplier bracket expansion	at most 200 doublings
Bisection stopping predicate	relative bracket width
Shared expert	excluded from the allocation
Router	never retrained or modified
Calibration split	`train` of `wikitext-2-raw-v1`

cold-start cost and scales with $\sum_l |\mathcal{C}_l|$ candidate evaluations, each dominated by the residual decompositions of that layer's E_l experts. Dominance pruning then reduces each layer to its nondominated frontier, after which a single multiplier query costs $\mathcal{O}(\sum_l |\mathcal{C}_l|)$ work, or less when the frontiers are sorted and breakpoints are reused across queries.

The multiplier search is a bisection on λ whose selected total rate is monotone (Appendix A.3), so each step halves the multiplier bracket and the number of steps is bounded, once the bracket expansion has fixed a positive lower multiplier, by the requested multiplier tolerance rather than by the size of the discrete grid. Writing $|\mathcal{C}| = \max_l |\mathcal{C}_l|$, the per-query cost $\mathcal{O}(\sum_l |\mathcal{C}_l|)$ above is at most $\mathcal{O}(L\,|\mathcal{C}|)$. The bisection is preceded by a bracket expansion that grows the upper multiplier from its start value until the selected rate fits the budget, and each probe there is one more query at the same cost, with a count that follows the budget rather than the tolerance. The allocation phase therefore costs $\mathcal{O}(L\,|\mathcal{C}|)$ multiplied by the expansion probes plus the bisection steps, the second of which the requested multiplier tolerance bounds under the same condition. The allocator tracks the feasible configuration produced at each visited multiplier and returns the lowest-distortion feasible one, so the returned point lies on the supported frontier; a budget that falls between supported points leaves the normalized slack of Eq. 8. The realization pass applies the chosen base operator and residual truncation to each layer exactly once. Wall-clock measurements for construction and allocation on the reported instances appear in Sec. 5, collected on the single NVIDIA A100 80 GB GPU under the stack of Appendix A.10.

Appendix A.12: When Joint Allocation Pays Off

This section consolidates the conditions under which the joint surface is worth its extra bookkeeping. Three readings of the reported comparison point the same way. In the main comparison, the

perplexity gap between the stage-wise baseline and the joint allocator widens as compression tightens, being smallest at the mildest ratio and largest at the most aggressive one. In the simple-allocator controls, a greedy allocator sits close to the joint allocator at mild compression and falls further behind as the budget tightens, while a uniform control degrades fastest throughout. In the factorial sweep, removing the joint coupling costs more than removing routing-weighted emphasis or changing materialization order; that sweep is evaluated at the same 40% Qwen operating point as Table S4.

Read together, these are consistent with a single mechanism: at aggressive ratios the shared base is pruned harder, which reshapes every residual spectrum more, so a decision rule that fixes the base before choosing ranks commits rate the joint surface could have placed elsewhere. For practitioners this gives a decision rule: the joint machinery earns its extra bookkeeping as compression tightens and the shared base is pruned harder.

APPENDIX A.13: ADDITIONAL EXPERIMENTAL DETAIL

APPENDIX A.13.1: FACTORIAL CELLS IN FULL

All eight cells of Table S7 are listed here, as perplexity and aggregate score, starting from the joint/routing/fixed corner at 8.91 and 55.71. Reversing materialization alone gives 8.97 and 55.64. Replacing routing weights alone gives 9.13 and 55.39, and reversing materialization as well gives 9.18 and 55.30. Separating the base and residual decisions alone gives 9.21 and 55.28, and reversing materialization as well gives 9.24 and 55.23. Separating the decisions and replacing routing weights gives 9.37 and 55.02, and combining all three alternatives reaches 9.43 and 54.91. The independent surface replaces $D_l(s, r)$ by $\tilde{D}_l(s, r) = D_l(s, r_0) + D_l(s_0, r) - D_l(s_0, r_0)$ at the smallest grid point $(s_0, r_0) = (0, 0)$ and removes dominated points again. By Eq. S1, $D_l - \tilde{D}_l$ is the mixed difference with corner (s_0, r_0); the independent arm therefore keeps both one-at-a-time effects exactly and sets the base–residual interaction to zero, on the same grids, rates, routing weights, and budget. Because the design is a complete factorial at one matched budget, the three main effects are separable; the interaction contrasts are determined as point estimates by the same eight cells, as computed in Appendix A.16.

What the two curves hold fixed. The Qwen rows of Table S4 form a complete compression curve within one model, so comparisons along that curve hold the backbone fixed and vary only the budget. The Mixtral rows repeat that curve on a second backbone, holding the family fixed and varying only the budget.

What the solver comparison is measured against. Solver wall times are measured on the bounded instance where exact dynamic programming is admitted, so the speedups are ratios against the exact oracle on that instance. EvoESAP, the one non-exact comparator scored on the same instance, sits at $(8.93 - 8.82)/8.82 = 0.0125$ in relative perplexity against the exact allocation, where RDO-MoE sits at 0.0057; the multiplier path therefore gives up 45% of what that search gives up while running $4.85\times$ faster than it. On this instance the returned allocation is compared with the exact one in perplexity, at 0.05, and in allocation time, at $19.6\times$.

APPENDIX A.14: BROADER IMPACTS AND RESPONSIBLE USE

This work concerns efficiency-oriented compression of existing public mixture-of-experts language models; it introduces no new model, no new training data, and no new deployment. The intended impact is to lower the storage and memory footprint of already-trained sparse models so that they can run below the datacenter tier, which can broaden access to capable open models on commodity accelerators.

Compression that improves an aggregate score can still degrade specific behaviors: the taskwise diagnostic shows per-task regressions at higher compression even where the aggregate improves, on a set of tasks that differs between the two backbones. A practitioner selecting an operating point should therefore consult the signed taskwise outcomes for the capabilities their application depends on, not the aggregate alone. The evaluation uses public benchmarks listed in Appendix A.6; no human-subjects data is collected and no new high-risk capability is released.

APPENDIX A.15: EXTENDED EXPERIMENTAL RESULTS

APPENDIX A.15.1: RESULT TABLES IN FULL

The result tables are collected here so that this appendix carries them on its own. Those that also appear in Sec. 5 are restated; the remainder are reported only here.

Table S4 is the compression-quality comparison. RDO-MoE holds the lowest perplexity and the highest aggregate score at all four Qwen levels, and the perplexity margin over D-Rank widens as the budget tightens; on Mixtral RDO-MoE again has the lowest perplexity while the task aggregate holds the four methods within one point.

Table S7: Restated from the main paper. Complete factorial at 40% Qwen compression. J/I: joint/independent surface; P/U: routing/uniform weights; S/R: fixed/reversed materialization.

Config.	PPL	Score	Config.	PPL	Score
J/P/S	8.91	55.71	J/P/R	8.97	55.64
J/U/S	9.13	55.39	J/U/R	9.18	55.30
I/P/S	9.21	55.28	I/P/R	9.24	55.23
I/U/S	9.37	55.02	I/U/R	9.43	54.91

Table S7 is the complete factorial. The joint, routing-weighted, fixed-materialization corner is the favorable one, and among the three axes the coupling axis carries the largest single-cell effect, 0.30 perplexity.

Table S8: Restated from the main paper. Solver comparison on the bounded 40% Qwen instance that exact DP can enumerate. Perplexities and wall times are for that restricted instance: RDO-MoE reads 8.87 and 8.12 s here against 8.91 and 9.65 s on the full model.

Method	PPL	Time (s)	Relative speed
Exact DP	8.82	159.4	1.0×
EvoESAP	8.93	39.4	4.0×
RDO-MoE	8.87	8.12	19.6×

Table S8 is the solver comparison on the bounded instance whose state space exact dynamic programming can enumerate. The multiplier sweep reaches 8.87 against the exact 8.82 in 8.12 seconds against 159.4, so the relaxation costs 0.05 perplexity for a $19.6\times$ reduction in solve time.

Table S9: Per-task scores at 40% Qwen compression. The unweighted mean of the nine accuracy-scored entries gives the 40% Qwen aggregate of Table S4. The two generation metrics sit right of the rule and are excluded from that aggregate. Task abbreviations: ARC-E, ARC-Easy; ARC-C, ARC-Challenge; Hella, HellaSwag; Wino, WinoGrande; OBQA, OpenBookQA; Trivia, TriviaQA; NQ, Natural Questions; HE, HumanEval.

	Accuracy-scored tasks (aggregate)									Generation	
Method	ARC-E	ARC-C	Hella	PIQA	Wino	BoolQ	OBQA	MMLU	Trivia	NQ	HE
D-Rank	66.7	43.4	60.9	75.8	61.2	65.0	34.8	51.6	39.3	13.6	19.7
RDO-MoE	67.7	44.2	62.0	76.4	60.3	63.7	35.2	52.1	39.8	13.8	20.9

Table S11 gives the signed taskwise differences behind those aggregates. Nine of eleven tasks stay favorable at both compression levels, while WinoGrande and BoolQ move against the aggregate and worsen as compression tightens, which is the disconnect the signed readout exists to expose.

APPENDIX A.15.2: ROBUSTNESS AND CALIBRATION

Sparsity-grid, rank-grid, and tolerance sensitivity. Table S13 shows that finer sparsity and rank grids improve quality only slightly after the default setting, while every refinement past it costs time. Quality is nearly flat from the default setting onward and time is monotone in resolution. The

Table S10: Per-task scores at 40% Mixtral compression. The unweighted mean of each row's nine entries gives $572.7/9 = 63.63$ for D-Rank and $576.5/9 = 64.06$ for RDO-MoE, which are the 40% Mixtral entries of Table S4 at the precision reported there. RDO-MoE is ahead on six of the nine and behind on three, so the aggregate gap is not a uniform shift.

Method	ARC-E	ARC-C	Hella	PIQA	Wino	BoolQ	OBQA	MMLU	Trivia
D-Rank	77.7	46.9	57.4	81.9	73.1	78.4	30.5	62.2	64.6
RDO-MoE	78.5	48.5	57.3	81.4	73.9	78.6	31.1	61.8	65.4

Table S11: Restated from the main paper. Taskwise score differences between RDO-MoE and D-Rank on Qwen; positive is favorable. Per-task absolute scores at 40% are in Table S9. Task abbreviations: ARC-E, ARC-Easy; ARC-C, ARC-Challenge; Hella, HellaSwag; Wino, WinoGrande; OBQA, OpenBookQA; Trivia, TriviaQA; NQ, Natural Questions; HE, HumanEval.

Compression	ARC-E	ARC-C	Hella	PIQA	Wino	BoolQ	OBQA	MMLU	Trivia	NQ	HE
40%	1.0	.8	1.1	.6	-.9	-1.3	.4	.5	.5	.2	1.2
60%	1.3	.9	1.4	.8	-1.5	-2.1	.5	.7	.4	.3	1.5

Table S12: Construction and allocation wall time, memory and throughput. The times are wall-clock on one A100 80 GB. Memory and throughput are ratios to the uncompressed model at the two Qwen points.

Model	Compression	Construction (s)	Allocation (s)	Memory ratio	Throughput ratio
Qwen	40%	155	9.65	.62	1.18
Qwen	60%	157	10.42	.43	1.27
Mixtral	40%	290	13.02	–	–

mechanism is diminishing returns near the frontier. Once the grid resolves the supported points the allocator selects, extra candidates and tighter stopping add search cost while moving perplexity by at most 0.02.

Table S13: Grid and tolerance sensitivity at 40% Qwen compression. Lower perplexity and wall time and higher score are favorable.

Condition	PPL	Score	Time (s)
Sparsity grid 3	9.04	55.3	9.38
Sparsity grid 7	8.91	55.7	9.65
Sparsity grid 15	8.89	55.8	10.21
Rank grid 4	9.07	55.2	9.32
Rank grid 8	8.91	55.7	9.65
Rank grid 16	8.90	55.7	10.34
Tolerance 10^{-2}	8.93	55.6	7.41
Tolerance 10^{-3}	8.91	55.7	9.65
Tolerance 10^{-4}	8.91	55.7	11.93

Figure S5 plots the same band across every grid condition, with the asterisk marking the default setting on each axis.

Routing, calibration, and resample sensitivity. Routing and calibration sensitivity in Table S14 supplies a distinct robustness check. The uniform-routing comparison changes the 40% point from 8.91/55.71 to 9.13/55.39, while the held-out routing response is 8.96/55.63. Across five resamples, perplexity has mean 8.914, standard deviation 0.011, and 95% half-width 0.014 over the evaluation set; the result is stable across calibration draws. Replacing routing-weighted expert emphasis with uniform weights changes what the objective treats as important, and therefore moves both perplexity and score together.

Table S14: Routing, calibration and resample sensitivity at 40% Qwen compression. The last three perplexity rows are the mean, sample standard deviation and confidence half-width over five resamples.

Condition	PPL	Score
Pre-compression routing	8.91	55.71
Uniform routing	9.13	55.39
Held-out routing	8.96	55.63
Five-resample mean	8.914 ± 0.014	n/a
Five-resample standard deviation	0.011	n/a
95% half-width	0.014	n/a

APPENDIX A.15.3: EFFICIENCY

Table S12 reports construction and allocation wall-clock together with memory and throughput, with the build and allocation times taken on a single NVIDIA A100 80 GB GPU under CUDA 12.8 and PyTorch 2.11 in bfloat16. For Qwen, construction is 155 seconds and allocation 9.65 seconds at 40%, against 157 and 10.42 seconds at 60%; memory relative to the uncompressed model is 0.62 at 40% compression and 0.43 at 60%, with throughput ratios of 1.18 and 1.27. Construction precedes the budget and is paid once per run as the cold-start cost of Appendix A.11, and the two build times therefore differ only by run-to-run wall-clock variation. The solver comparison of Table S8 times the allocation step alone. The allocation time is the higher of the two at 60%, where the bracket expansion runs at least as many probes before the bisection starts: the multiplier is doubled from its start value until the selected rate fits, each probe is one pass over the layer surfaces, and a tighter budget needs a multiplier at least as large before it does. The bisection that follows stops on bracket width relative to the upper multiplier, so once the bracket is expanded its step count is bounded by the tolerance. Mixtral uses a 290-second construction and a 13.02-second allocation at the 40% point. Each query is linear in depth at a fixed grid, and Mixtral-8×7B has $L = 32$ layers against Qwen's $L = 24$ at the same $|\mathcal{C}| = 56$; its allocation is the longer of the two by roughly the depth ratio $32/24$: $9.65 \times (32/24) = 12.87$ against the 13.02 reported, a 1.2% excess over the depth-scaled prediction. On Qwen, memory and throughput improve together from 40% to 60% compression.

APPENDIX A.15.4: GENERALIZATION AND MECHANISM ASSOCIATIONS

Generalization has two levels: the Qwen curve compares budgets within one model, while the Mixtral rows of Table S4 repeat the comparison across model families. Both favor the joint allocator on perplexity, while the Mixtral task aggregate holds the four methods within one point.

Table S15: Association between reconstruction-based configuration ranking and task ranking over 56 joint configurations. Task abbreviations: ARC-E, ARC-Easy; ARC-C, ARC-Challenge; Hella, HellaSwag; Wino, WinoGrande; OBQA, OpenBookQA; Trivia, TriviaQA; NQ, Natural Questions; HE, HumanEval.

Task	ARC-E	ARC-C	Hella	PIQA	HE	OBQA	MMLU	Trivia	NQ	Wino	BoolQ
Kendall τ	.81	.78	.79	.76	.71	.62	.58	.51	.47	.20	.00

Table S15 compares the reported reconstruction-based configuration order with taskwise order over 56 joint configurations. These 56 are the $|\mathcal{S}| \times |\mathcal{R}_0| = 7 \times 8$ enumerated joint grid points under the shared-per-layer-rank construction, evaluated over the admitted sparsity and rank grids. The eleven-task mean is $\tau = 0.57$, the value annotated in the right panel of Figure 2, and the spread around it is what makes the mean an unsafe summary on its own. That spread shows that a reconstruction proxy can rank configurations well on some tasks and not at all on others. The pattern is consistent with reconstruction preserving likelihood-aligned surface features but not the structure some answers depend on; on Qwen, BoolQ at $\tau = .00$ is also the deepest negative task in Table S11.

Table S16: Simple-allocator controls at all four Qwen compression levels, reported as differences from RDO-MoE: a uniform control, a greedy allocator, and a sequential D2-MoE comparison. The uniform and D2-MoE rows are arithmetic differences against Table S4; all four compression levels follow the same matched-budget protocol.

	30%		40%		50%		60%	
Method	ΔPPL	ΔScore	ΔPPL	ΔScore	ΔPPL	ΔScore	ΔPPL	ΔScore
Uniform	1.20	-2.3	1.93	-2.8	3.12	-3.1	5.79	-3.5
Greedy	.072	-.18	.137	-.27	.203	-.31	.318	-.42
D2-MoE	.20	-.30	.36	-.50	.52	-.40	.88	-.50

Appendix A.15.5: Simple-Allocator Controls

Table S16 provides the complementary matched-budget response. Greedy allocation is close at 30% (0.072 perplexity and 0.18 score points behind RDO-MoE), while the response reaches a 0.318 perplexity difference at 60%. Uniform allocation deteriorates much faster, from 1.20 to 5.79 perplexity points. D2-MoE lies between these controls. The mechanism consistent with this widening is that greedy allocation captures most of the frontier when interaction is mild. As compression tightens, the base and residual decisions become jointly constrained, and a per-step choice then commits rate the joint surface would have placed elsewhere. At 30% the greedy gap of 0.072 perplexity is five times the resample half-width measured at 40%.

On a relative scale the greedy response is 0.85%, 1.54%, 2.04% and 2.64% of the RDO-MoE perplexity at 30, 40, 50 and 60% compression. Set against the stage-wise deltas of the same matched-budget table, the greedy deficit is 36 to 39% of the stage-wise one at every level.

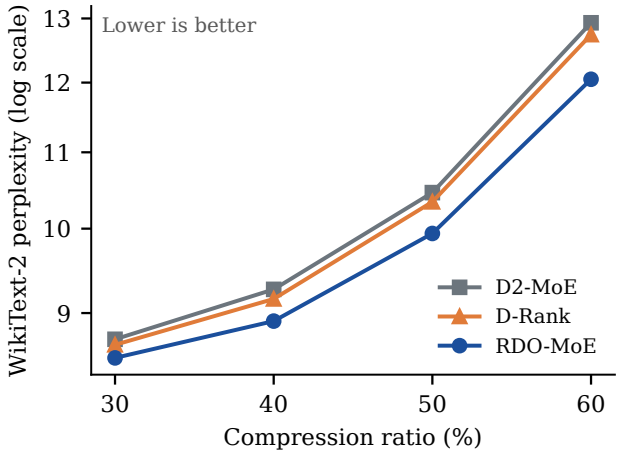


Figure S3: Qwen1.5-MoE-A2.7B WikiText-2 perplexity versus compression ratio (log vertical axis; lower is better), plotting the D2-MoE, D-Rank, and RDO-MoE rows of the Qwen block of Table S4.

Summary of the extended results. The analyses above move from the quality curve and its task-wise decomposition to the solver comparison, the resource and robustness profiles, and the simple-allocator controls; Table S5 in Appendix A.10 maps each reconstruction item to the section that fixes it.

Figure S3 plots the Qwen progression across the full compression range: the joint and rank-only allocations stay close at mild ratios yet separate more visibly at the tightest one, whereas the stage-wise delta-decomposition comparison falls away fastest of the three plotted series.

Figure S4 makes the taskwise pattern visible: RDO-MoE leads on nine of eleven tasks while trailing on the two starred negative tasks. Table S10 gives the same decomposition at the Mixtral 40% point, where the split is six ahead and three behind, on HellaSwag, PIQA and MMLU. The figure separates

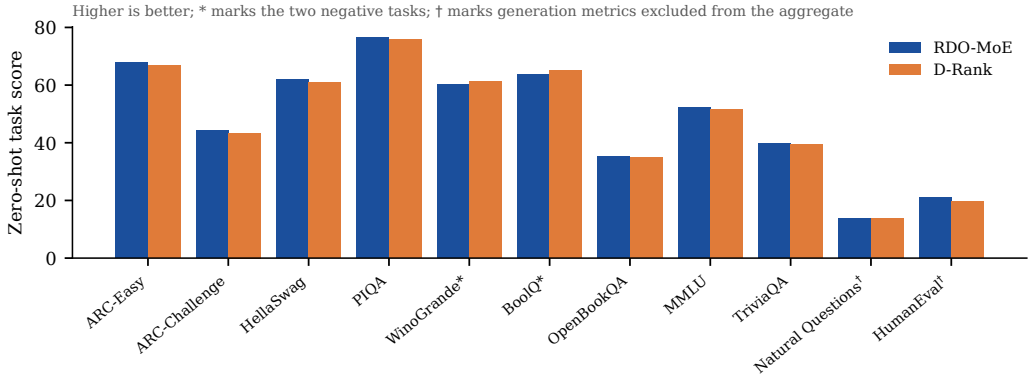


Figure S4: Per-task zero-shot scores at 40% Qwen compression. WinoGrande and BoolQ (starred) are the only tasks where RDO-MoE falls below D-Rank; the daggered bars are the generation metrics, excluded from the aggregate.

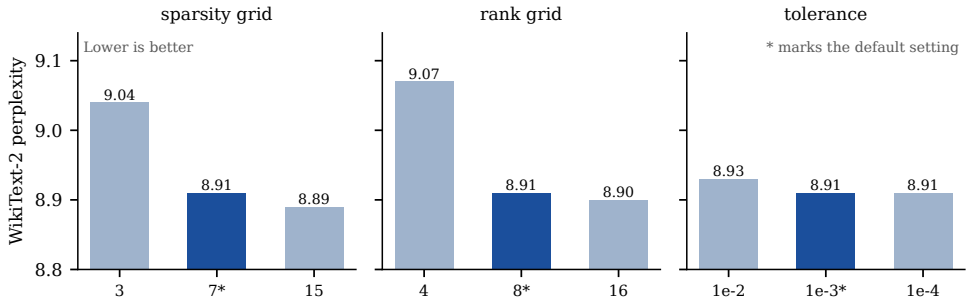


Figure S5: Perplexity sensitivity across the sparsity-grid, rank-grid and tolerance settings at 40% Qwen compression; the asterisk marks each default. The spread stays within 8.89 to 9.07.

the two generation metrics from the aggregate. Natural Questions and HumanEval sit near 14 and 20 points while the accuracy-scored tasks span roughly 35 to 76. Pooling them would compress the summary toward a value that no single task family attains.

Appendix A.16: Interaction Contrasts of the Factorial Design

The complete factorial of Table S7 crosses three binary choices at one matched 40% Qwen budget: joint versus independent surface (J/I), routing-weighted versus uniform expert weights (P/U), and fixed versus reversed materialization (S/R). The design is complete, so every two-factor and three-factor contrast is identified at this budget.

Write $y(\cdot)$ for a cell of that table. For factors X and Y, the two-factor contrast averaged over the remaining factor Z is

$$\widehat{\Gamma}_{XY} = \tfrac{1}{2} \sum_{z} \big[y(X_2, Y_2, z) + y(X_1, Y_1, z) - y(X_2, Y_1, z) - y(X_1, Y_2, z) \big], \tag{S7}$$

the difference between the effect of X at one level of Y and its effect at the other. The three-factor contrast is half the difference of the two two-factor contrasts formed at the two levels of Z, on the same scale. Main effects are reported as differences of the four-cell means, which average over the two levels of each remaining factor and so differ from the single-cell drops read off the favorable corner in Appendix A.13.1.

Two readings follow from Table S17. First, the largest two-factor contrast is 0.040 in perplexity against a largest main effect of 0.265, a ratio near one to seven, and the two remaining two-factor

Table S17: Interaction and main-effect contrasts of the factorial design of Appendix A.13.1, by Eq. S7. Perplexity and aggregate score are on the scales of that table; main effects are differences of the four-cell means. Every entry is a contrast of the same eight cells.

Quantity	PPL	Score
Surface × weights (J/I × P/U)	-0.040	$+0.040$
Surface × materialization (J/I × S/R)	-0.010	0.000
Weights × materialization (P/U × S/R)	$+0.010$	-0.040
Surface × weights × materialization	$+0.020$	-0.020
Main effect, surface (I minus J)	$+0.265$	-0.400
Main effect, weights (U minus P)	$+0.195$	-0.310
Main effect, materialization (R minus S)	$+0.050$	-0.080

contrasts are a further factor of four smaller. The ordering reported in Sec. 5 is therefore a statement about main effects, with the design surface close to additive over these three axes at this budget. Second, the cells carry two decimals, so each contributes at most 0.005 of rounding and a contrast built from eight cells with coefficient $\frac{1}{2}$ inherits at most 0.020. In perplexity the surface-by-weights contrast at 0.040 is twice that bound, the two remaining two-factor contrasts sit inside it, and the three-factor contrast sits at it; in score both the surface-by-weights and the weights-by-materialization contrasts reach 0.040. The sign of the surface-by-weights contrast in perplexity, negative, says the penalty for separating the base and residual decisions is smaller when expert weights are already uniform than when they are routing-weighted, so their perplexity effects are not exactly additive.

These quantities are contrasts in perplexity and aggregate score over three design axes. The mixed difference Γ_l of Appendix A.3 is a different object, defined on the distortion objective, per layer, over the sparsity and rank grids. Both measure departure from an additive form over two crossed choices. They differ in the space measured, the resolution of the grid, and the aggregation over layers, so a nonzero entry above indicates departure from additivity on the design surface rather than a value of Γ_l itself. A complete factorial at one matched budget fixes each contrast as a point estimate.