HARP: HIERARCHICAL ADAPTIVE REDISTRIBUTION FOR PRUNING MIXTURE-OF-EXPERTS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Channel-level pruning can reduce both the parameter footprint and activated computation of mixture-of-experts (MoE) models, but doing so effectively requires deciding how a fixed global channel budget should be distributed across layers and experts, as well as which channels should be retained. Existing approaches often rely on calibration activations, whose quality can vary substantially across routed experts and calibration distributions. We present HARP, a calibration-free framework for hierarchical adaptive redistribution of MoE channel capacity. HARP derives a model-intrinsic structural preservation (SP) prior from the relative fluctuation of retained absolute-weight mass under random thinning and applies the same functional at the layer, expert, and coupled-channel levels. A hierarchical allocator uses these priorities to construct hardware-aligned heterogeneous expert widths under an exact global budget, while Channel-SP realizes each width through coupled channel selection. When calibration data are available, empirical criteria can refine channel identities without changing the global allocation. Across Qwen3, Qwen3.6, and DeepSeek-V2-Lite at 25% and 50% channel pruning, HARP achieves the highest mean retained performance among the evaluated channel-pruning methods in all six settings, consistently outperforming calibration-based Wanda and ENP. SP anchoring improves calibration-assisted channel selection and reduces calibration sensitivity. Finally, a width-aware vLLM backend executes heterogeneous experts at their retained widths, achieving 1.04–$1.54\times$ end-to-end inference speedup on a single A100 GPU.

1 INTRODUCTION

Mixture-of-Experts (MoE) models scale parameter capacity through sparse conditional computation, activating only a small subset of experts for each token. Sparse routing, however, does not remove the deployment cost of the expert pool: all experts must still be stored, while the selected experts remain responsible for a substantial fraction of inference computation. Coarse-grained layer or expert pruning removes complete computational units, whereas channel pruning reduces individual expert widths while preserving the pretrained router, expert identities, and top-k routing topology (Xie et al., 2024; He et al., 2026). This finer granularity reduces both stored parameters and activated computation while allowing different layers and experts to retain different amounts of capacity. Recent heterogeneous MoE pruning results show that such non-uniform allocation can outperform uniform pruning, especially at high compression ratios. Channel pruning therefore provides a natural mechanism for redistributing a fixed capacity budget throughout the model.

Exploiting this flexibility raises the first challenge: where should the remaining capacity be placed? A fixed global channel budget must specify layer totals, expert widths, and the channels that realize those widths. These decisions are coupled: local channel rankings cannot determine cross-layer or cross-expert allocation, and a poor coarse allocation cannot be repaired downstream. Existing heterogeneous approaches obtain their allocation from prescribed structural patterns or calibration-dependent estimates. What remains missing is a model-adaptive mechanism that converts one global budget into layer-, expert-, and channel-level preservation decisions.

A second challenge concerns what evidence should determine preservation. Many strong pruning criteria estimate channel importance from activations on a calibration corpus, but the routed experts are not observed equally. Across four calibration corpora, 30–81 Qwen3 layer–expert slots receive

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.

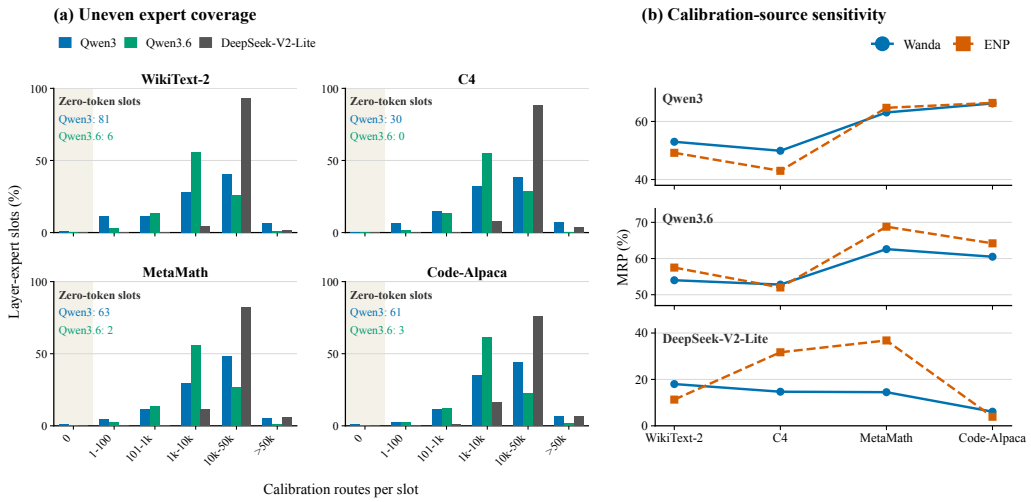

Figure 1: Calibration-based channel importance rests on uneven expert coverage and changes with calibration data. (a) Percentage of layer–expert slots in each route-count bin for four calibration corpora, using Wanda routing statistics and normalizing within each model. (b) Mean retained performance (MRP) of uniform-width Wanda and ENP at 50% pruning. Within each model–method trajectory, only the calibration corpus changes; y-axis scales differ across models.

no calibration tokens, while 325–454 receive more than 50,000 routes (Fig. 1a). Activation-derived importance is estimated from markedly unequal evidence across experts. The evidence depends on the calibration distribution: with model, method, architecture, and sparsity held fixed, changing only the corpus alters downstream retained performance for uniform-width Wanda and ENP (Fig. 1b). The two criteria need not even change in the same direction. Calibration evidence remains useful, but it cannot be treated as an invariant measure of structural importance. This empirical challenge complements the allocation problem: MoE channel pruning must decide both where to place capacity and how to use distribution-dependent evidence when selecting channels.

We introduce HARP, a model-intrinsic framework that addresses both challenges through a structural preservation (SP) prior from pretrained weights. SP interprets weight concentration through the relative fluctuation of retained parameter mass under random thinning and assigns priorities independently at the layer, expert, and coupled-channel levels. A hierarchical allocator converts these priorities into hardware-aligned heterogeneous expert widths under an fixed global budget, while Channel-SP selects the retained channels. The default pipeline requires no calibration, fine-tuning, or candidate-model evaluation. When calibration data are available, SP anchors Wanda or ENP rankings so that empirical evidence refines local selection without determining the allocation itself.

Across Qwen3, Qwen3.6, and DeepSeek-V2-Lite at 25% and 50% channel pruning, HARP achieves the highest mean retained performance among channel-pruning methods in all six settings. At 50% pruning on Qwen3.6, it retains 86.0% of dense performance, compared with four-corpus averages of 57.5% for Wanda and 60.6% for ENP. Ablations isolate gains from hierarchical redistribution and structural channel preservation. With identical widths and calibration data, SP anchoring improves Wanda or ENP selection in 47 of 48 comparisons. A width-aware vLLM implementation further converts heterogeneous widths into 1.04–1.54$\times$ end-to-end speedup on one A100 GPU.

Our contributions are threefold:

- **Hierarchical MoE capacity allocation.** We formulate channel pruning as an exact global-budget allocation problem across layers, experts, and coupled channels, producing model-adaptive heterogeneous widths rather than uniform or predefined sparsity patterns.
- **A model-intrinsic structural preservation prior.** We derive preservation priorities from pretrained weights for calibration-free pruning and use the same prior to anchor calibration-derived channel criteria when empirical evidence is available.

- **Executable and efficient heterogeneous compression.** Across three MoE families and two pruning ratios, HARP achieves the strongest channel-pruning performance, while a true-width vLLM backend translates its heterogeneous expert widths into end-to-end acceleration and improved performance–efficiency trade-offs over uniform-width pruning.

2 Related Work

2.1 MoE Pruning Across Structural Granularities

MoE compression operates at different structural granularities. Whole-expert pruning removes complete experts using routing, activation, reconstruction, or weight-based importance estimates. AIMER, for example, derives a calibration-free weight-concentration criterion for task-agnostic expert ranking (Liu et al., 2026a). While such methods reduce the resident expert pool, they do not proportionally reduce the dominant MLP computation of each routed expert when surviving experts retain their original widths. Intra-expert structured pruning instead reduces routed-expert dimensions while preserving the expert set and routing topology. MoE-Pruner combines router- and activation-aware signals with weight pruning (Xie et al., 2024); TENP prunes expert neurons using calibration-derived projected contributions (He et al., 2026); and attribution-guided structural pruning combines channel scoring with coverage-aware allocation (Ding et al., 2026). HARP differs by deriving both heterogeneous expert widths and retained channels directly from pretrained weights, without modifying the expert set or routing mechanism.

2.2 Heterogeneous Capacity Allocation

Structured pruning allows the pruning budget to vary across layers and experts. In dense LLMs, OWL assigns non-uniform layerwise sparsity from outlier statistics (Yin et al., 2024), while EvoPress searches compression configurations under a global resource constraint (Sieberling et al., 2025). Recent MoE methods similarly exploit heterogeneous capacity: EvoESAP searches cross-layer expert budgets (Liu et al., 2026b), TENP uses a prescribed depth-dependent trapezoidal profile (He et al., 2026), and attribution-guided pruning allocates capacity using calibration-derived attribution and coverage (Ding et al., 2026). HARP instead formulates heterogeneity as a hierarchical allocation problem: a single weight-derived structural prior assigns layer, expert, and channel priorities under one global budget, jointly determining heterogeneous expert widths and retained channels without calibration. This differs from prescribed depth profiles, search-based configurations, or calibration-derived budgets, and adapts to pretrained weights.

2.3 Calibration-Dependent Importance and Structural Priors

Many post-training pruning criteria estimate importance from calibration activations. SparseGPT uses approximate second-order reconstruction statistics (Frantar & Alistarh, 2023), Wanda combines weight magnitude with activation norms (Sun et al., 2024b), and MoE-specific methods such as TENP likewise rely on calibration-dependent expert-neuron estimates (He et al., 2026). Such signals can capture input-dependent information, but their estimates depend on the calibration distribution. This sensitivity has been observed directly in LLM pruning: the choice of calibration corpus can substantially affect post-pruning performance, and C4 is not universally optimal (Bandari et al., 2024); more recent work further reports capability-specific trade-offs across general, mathematical, and code calibration sources (Xu et al., 2026). HARP addresses this variability with a model-intrinsic structural prior rather than a new calibration corpus. The prior supports calibration-free pruning and can also anchor Wanda or ENP channel rankings, allowing empirical evidence to refine local selection while the global allocation remains determined by pretrained weights.

3 Method

Overview. HARP consists of two components. Unified Structural Prior (USP) extracts model-intrinsic priorities at the layer, expert, and coupled-channel levels, and Hierarchical Capacity Allocation (HCA) realizes them as hardware-aligned expert widths and retained channel sets. Optional calibration evidence changes only channel selection. Figure 2 summarizes the framework.

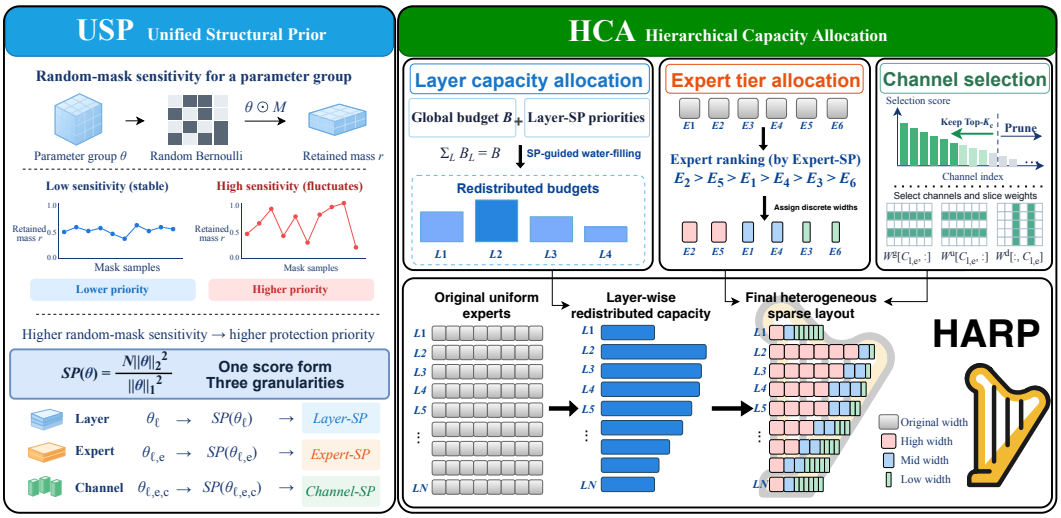

Figure 2: Overview of HARP. USP derives model-intrinsic preservation priorities from pretrained weights, and HCA realizes them as a heterogeneous routed-expert architecture under an fixed global budget. The channel-selection score is Channel-SP alone without calibration; with calibration, it sums within-expert ranks of Channel-SP and Wanda or ENP for observed experts, while unseen experts fall back to Channel-SP. Calibration leaves layer budgets and expert widths unchanged.

3.1 Problem Formulation

Consider an MoE model with L MoE layers and E routed experts per layer. Each expert has hidden dimension d and intermediate width D and computes

$$f_{\ell,e}(h) = W^d_{\ell,e}\left[\phi(W^g_{\ell,e}h) \odot (W^u_{\ell,e}h)\right], \tag{1}$$

where $W^g_{\ell,e}, W^u_{\ell,e} \in \mathbb{R}^{D\times d}$ and $W^d_{\ell,e} \in \mathbb{R}^{d\times D}$. Intermediate channel c couples the c-th gate and up rows with the c-th down-projection column, so these slices must be retained jointly. For a retained set $\mathcal{C}_{\ell,e} \subseteq \{1,\dots,D\}$ with width $K_{\ell,e} = |\mathcal{C}_{\ell,e}|$, the pruned expert uses

$$W^{g\prime}_{\ell,e} = W^g_{\ell,e}[\mathcal{C}_{\ell,e},:], \quad W^{u\prime}_{\ell,e} = W^u_{\ell,e}[\mathcal{C}_{\ell,e},:], \quad W^{d\prime}_{\ell,e} = W^d_{\ell,e}[:,\mathcal{C}_{\ell,e}]. \tag{2}$$

Bias terms, when present, follow the same indices. Given a global routed-expert channel budget B, the objective is to determine the widths and retained sets such that

$$\sum_{\ell=1}^{L}\sum_{e=1}^{E} K_{\ell,e} = B. \tag{3}$$

HARP modifies only routed-expert intermediate channels. Attention modules, router parameters, shared experts, non-MoE layers, the expert set, and the pretrained top-k routing mechanism remain unchanged.

3.2 Unified Structural Prior

USP measures how vulnerable the parameter mass of a structural unit is to partial removal. Let $\Theta = (\theta_i)_{i=1}^N \neq 0$ contain N scalar parameters, and independently retain each coordinate with $m_i \sim \text{Bernoulli}(q)$. The retained absolute mass $Z_q = \sum_i m_i|\theta_i|$ has

$$\mathbb{E}[Z_q] = q\|\Theta\|_1, \qquad \text{Var}(Z_q) = q(1-q)\|\Theta\|_2^2. \tag{4}$$

Let $\text{CV}(Z_q) = \sqrt{\text{Var}(Z_q)}/\mathbb{E}[Z_q]$ denote the coefficient of variation. Normalizing $\text{CV}^2(Z_q)$ by the equal-magnitude reference $(1-q)/(qN)$ removes both q and group dimensionality, yielding

$$\text{SP}(\Theta) = \frac{qN}{1-q}\,\text{CV}^2(Z_q) = \frac{N\|\Theta\|_2^2}{\|\Theta\|_1^2}. \tag{5}$$

A larger coefficient of variation indicates greater sensitivity to random-thinning perturbations; HARP therefore assigns the corresponding structural unit a higher preservation priority. The score is scale invariant and satisfies $1 \leq \mathrm{SP}(\Theta) \leq N$. The random-thinning construction provides the statistical interpretation; HARP deterministically allocates capacity and selects coupled channels. Appendix A gives the derivation and properties.

We instantiate the same functional independently at all three structural levels. Let Θ^L_ℓ contain the routed-expert parameters of layer ℓ, $\Theta^E_{\ell,e}$ contain the gate, up, and down weights of expert (ℓ, e), and $\Theta^C_{\ell,e,c}$ contain the three coupled slices of channel c. The resulting priorities are

$$S^L_\ell = \mathrm{SP}(\Theta^L_\ell), \qquad S^E_{\ell,e} = \mathrm{SP}(\Theta^E_{\ell,e}), \qquad S^C_{\ell,e,c} = \mathrm{SP}(\Theta^C_{\ell,e,c}). \tag{6}$$

Each priority is computed from its structural unit rather than by averaging finer-grained scores. The three levels therefore encode distinct information for cross-layer allocation, expert-width assignment, and coupled-channel selection.

3.3 Hierarchical Capacity Allocation

HCA converts the three priority collections into an executable architecture satisfying Equation (3). Each routed expert is restricted to three aligned widths,

$$\begin{aligned} &K_{\ell,e} \in \mathcal{K} = \{K_{\mathrm{low}}, K_{\mathrm{mid}}, K_{\mathrm{high}}\}, \\ &K_{\mathrm{mid}} = K_{\mathrm{low}} + \Delta, \qquad K_{\mathrm{high}} = K_{\mathrm{low}} + 2\Delta, \qquad B = LEK_{\mathrm{mid}}. \end{aligned} \tag{7}$$

HCA redistributes the $LE\Delta$ channels above the all-K_{low} configuration across layers and experts according to Layer-SP and Expert-SP, producing heterogeneous widths under the global budget.

Layer capacity allocation. For response exponent $\gamma > 0$, Layer-SP defines the normalized multiplier

$$a_\ell = \frac{L(S^L_\ell)^\gamma}{\sum_{j=1}^L (S^L_j)^\gamma}, \qquad \frac{1}{L}\sum_{\ell=1}^L a_\ell = 1. \tag{8}$$

Layer ℓ then receives the continuous per-expert target

$$K^{\mathrm{target}}_\ell = K_{\mathrm{low}} + a_\ell \Delta, \qquad \frac{1}{L}\sum_{\ell=1}^L K^{\mathrm{target}}_\ell = K_{\mathrm{mid}}. \tag{9}$$

The exponent γ controls the strength of cross-layer redistribution without changing the budget. If K^{target}_ℓ exceeds K_{high}, it is clipped to K_{high}, and the excess $\sum_\ell \max(0, K^{\mathrm{target}}_\ell - K_{\mathrm{high}})$ is redistributed to unsaturated layers in Layer-SP order until the total returns to LK_{mid}.

Expert-width realization. The continuous layer targets are projected onto discrete tier counts. Layers are processed in descending Layer-SP order. For layer ℓ, the available capacity is $B_\ell = EK^{\mathrm{target}}_\ell + r_{\ell-1}$, where $r_0 = 0$ and $r_{\ell-1}$ carries unused capacity from preceding allocations. If (n_l, n_m, n_h) are the numbers of low-, middle-, and high-width experts, their utilization is $U_\ell = n_l K_{\mathrm{low}} + n_m K_{\mathrm{mid}} + n_h K_{\mathrm{high}}$. HCA selects

$$(n^\star_l, n^\star_m, n^\star_h) \in \underset{n_l, n_m, n_h \in \mathbb{Z}_{\geq 0}}{\arg\max} \; U_\ell, \quad \text{s.t.} \quad n_l + n_m + n_h = E, \quad U_\ell \leq B_\ell. \tag{10}$$

Whenever feasible, each tier additionally contains at least $\lceil \eta E \rceil$ experts, avoiding degenerate width buckets in data-free allocation. Expert-SP assigns K_{high} to the highest-ranked $n^\star_h$ experts, K_{mid} to the next $n^\star_m$, and K_{low} to the remainder. Unused capacity $r_\ell = B_\ell - U^\star_\ell$ is propagated to the next processed layer. After all layers, residual capacity is consumed by Layer-SP- and Expert-SP-prioritized single-tier upgrades. Because every upgrade adds exactly Δ channels, this closure reaches the global budget exactly.

Channel realization. Given the realized width, HCA retains the highest-priority coupled channels:

$$\mathcal{C}_{\ell,e} = \mathrm{TopK}_c\big(S^C_{\ell,e,c}, K_{\ell,e}\big)\,. \tag{11}$$

The selected indices are applied jointly according to Equation (2). Here, γ controls cross-layer redistribution, η the minimum population of each tier, and Δ the spacing between aligned widths.

3.4 Calibration-Assisted Channel Selection

When calibration data are available, empirical criteria can refine channel identities without changing Layer-SP allocation, Expert-SP ordering, realized widths, or the global budget. Let $A_{\ell,e,c}$ be an empirical criterion such as a Wanda- or ENP-style score. We orient both criteria so that larger values indicate stronger preservation priority and rank-normalize them within each expert: $z^{\mathrm{SP}}_{\ell,e,c} = \mathcal{R}_{\ell,e}(S^C_{\ell,e,c})$ and $z^{\mathrm{Emp}}_{\ell,e,c} = \mathcal{R}_{\ell,e}(A_{\ell,e,c})$, where $\mathcal{R}_{\ell,e}$ maps ranks to $[0,1]$. Let $N_{\ell,e}$ be the number of calibration tokens routed to expert (ℓ,e), so that $\sum_e N_{\ell,e} = k|\mathcal{D}_{\mathrm{cal}}|$. The anchored score and retained set are

$$\widetilde{S}^C_{\ell,e,c} = z^{\mathrm{SP}}_{\ell,e,c} + \mathbf{1}[N_{\ell,e} > 0] z^{\mathrm{Emp}}_{\ell,e,c}, \qquad \mathcal{C}^{\mathrm{Emp}}_{\ell,e} = \mathrm{TopK}_c\Big(\widetilde{S}^C_{\ell,e,c}, K_{\ell,e}\Big). \tag{12}$$

The indicator suppresses the empirical term for unseen experts, so the score reduces to Channel-SP when $N_{\ell,e} = 0$. We deliberately keep this combination simple: since calibration evidence is uneven across experts and varies with the corpus, a fixed equal-weight rank sum prevents the empirical term from dominating the structural prior, while still allowing local refinement. This separation lets model-intrinsic structure determine where capacity is placed while distribution-dependent evidence refines which channels realize that capacity.

3.5 True-Width Execution

The heterogeneous architecture reduces computation only when experts execute at their retained widths rather than being padded to the maximum tier width K_{high}. Our width-aware backend groups experts by their tier $K_{\ell,e}$ and dispatches each group to width-specific computation over the retained gate/up rows and down-projection columns. The gate, up, and down projections each cost $O(dK_{\ell,e})$, so the dominant projection cost for a token routed to expert (ℓ,e) scales as $3dK_{\ell,e}$. With top-k routing, the per-token cost is $3d\sum_{e\in\mathrm{top}-k} K_{\ell,e}$. Reducing expert width therefore directly reduces arithmetic whenever that expert is selected, while preserving the pretrained routing behavior.

4 Experiments

4.1 Evaluation protocol

We evaluate Qwen3-30B-A3B-Instruct-2507, Qwen3.6-35B-A3B, and DeepSeek-V2-Lite-Chat at 25% and 50% pruning (Yang et al., 2025; DeepSeek-AI et al., 2024). The eight-task suite contains ARC, HellaSwag, WinoGrande, GSM8K, MATH-500, MMLU, HumanEval, and MBPP (Clark et al., 2018; Zellers et al., 2019; Sakaguchi et al., 2020; Cobbe et al., 2021; Hendrycks et al., 2021b;a; Chen et al., 2021; Austin et al., 2021). Mean retained performance (MRP) is the unweighted mean percentage retained relative to the dense checkpoint. The channel baselines are random selection, structured Wanda (Sun et al., 2024a), and ENP from TENP (He et al., 2026). Wanda and ENP use WikiText-2 (Merity et al., 2017), C4, MetaMath, and Code-Alpaca calibration; Table 1 reports their four-corpus average. Each protocol uses 128 sequences of length 2048, with rankings frozen before evaluation. We use vLLM (Kwon et al., 2023) and EvalScope (ModelScope Team, 2024) with greedy decoding, seed 42, thinking disabled, and complete splits. Unless stated otherwise, HARP uses $(\gamma,\eta,\Delta) = (2, 0.15, 128)$ throughout. Appendix C gives the complete protocol.

4.2 Overall performance and hierarchical capacity allocation

Overall performance. Table 1 compares all methods under a common routed-expert storage budget and reports mean retained performance (MRP) together with per-token routed-FFN computation relative to the dense model. HARP achieves the highest MRP among all evaluated channel-pruning methods in all six model–sparsity settings, consistently outperforming both calibration-based Wanda and ENP and the calibration-free Random and CSP baselines. The advantage is particularly pronounced at 50% pruning, where HARP reaches 69.3, 86.0, and 38.2 MRP on Qwen3, Qwen3.6, and DeepSeek-V2-Lite, compared with 59.0, 73.9, and 37.2 for uniform-width CSP. The improvement is also consistent at 25% pruning, indicating that HARP remains effective across both moderate and aggressive compression regimes rather than relying on a single sparsity setting. Overall, these results show that the model-intrinsic structural prior generalizes across different MoE architectures and provides increasingly strong benefits as the retained channel budget becomes more constrained.

Benefits of hierarchical redistribution. CSP, HSP, and HARP form a controlled progression with increasingly flexible capacity allocation: CSP applies Channel-SP at uniform expert widths, HSP additionally assigns heterogeneous widths within each layer using Expert-SP, and HARP further redistributes capacity across layers using Layer-SP. This progression progressively relaxes the equal-capacity assumption, first across experts and then across layers, while retaining the same structural-prior principle for preservation. At 50% pruning, it yields 59.0→61.9→69.3 MRP on Qwen3 and 73.9→80.6→86.0 on Qwen3.6; at 25% pruning, the corresponding results are 91.9→93.1→94.8 and 94.0→95.9→97.1. These consistent improvements show that HARP benefits not only from identifying important channels, but also from determining how much capacity different experts and layers should retain. The larger margins at 50% pruning on both Qwen models further show that hierarchical redistribution becomes increasingly valuable when the global budget is tight and allocation mistakes are harder to absorb downstream.

Table 1: Overall pruning performance and routed-expert efficiency. Each cell gives MRP (relative per-token routed-FFN computation). Wanda and ENP report four-corpus averages. AIMER is an expert-level reference, and bold marks the highest channel-pruning MRP.

		Calibration-based		Data-free channel pruning				Expert ref.
Model	Pruning	Wanda	ENP	Random	CSP	HSP	HARP	AIMER
Qwen3	25%	88.7 (0.75×)	89.3 (0.75×)	87.8 (0.75×)	91.9 (0.75×)	93.1 (0.777×)	**94.8** (0.781×)	95.4 (1.00×)
	50%	58.1 (0.50×)	55.8 (0.50×)	46.1 (0.50×)	59.0 (0.50×)	61.9 (0.519×)	**69.3** (0.523×)	71.8 (1.00×)
Qwen3.6	25%	92.0 (0.75×)	91.8 (0.75×)	92.0 (0.75×)	94.0 (0.75×)	95.9 (0.776×)	**97.1** (0.778×)	97.6 (1.00×)
	50%	57.5 (0.50×)	60.6 (0.50×)	57.5 (0.50×)	73.9 (0.50×)	80.6 (0.526×)	**86.0** (0.529×)	72.4 (1.00×)
DeepSeek-V2-Lite	25%	15.0 (0.75×)	60.3 (0.75×)	61.7 (0.75×)	65.8 (0.75×)	68.3 (0.752×)	**69.5** (0.752×)	64.0 (1.00×)
	50%	13.3 (0.50×)	20.9 (0.50×)	0.2 (0.50×)	37.2 (0.50×)	37.0 (0.502×)	**38.2** (0.503×)	23.7 (1.00×)

Performance–compute efficiency. HARP's heterogeneous capacity allocation preserves more performance while keeping routed computation low. Compared with equal-width CSP under the same global channel-storage budget, HARP introduces only a small routing-induced increase in computation: at 50% pruning, routed-FFN computation changes from $0.500\times$ to $0.523\times$ on Qwen3 and $0.529\times$ on Qwen3.6, while MRP improves from 59.0 to 69.3 and from 73.9 to 86.0, respectively. On DeepSeek-V2-Lite, HARP reaches 38.2 MRP at $0.503\times$ computation, compared with 37.2 at $0.500\times$ for CSP. HARP is also competitive with the whole-expert pruning reference AIMER while using substantially less routed-FFN computation: HARP operates at approximately 0.50–$0.78\times$ dense routed-FFN computation across the evaluated settings, whereas AIMER retains $1.00\times$. By pruning expert widths rather than entire experts, HARP preserves the expert set and pretrained routing topology while reducing the computation of activated experts.

4.3 Structural score and allocator sensitivity

Table 2: Mechanism ablations on Qwen3.6 at 50% pruning. (a) Structural-score replacement. (b) One-at-a-time allocator sensitivity; bold marks the default setting.

(a) Score ablation

Variant	L	E	C	MRP
All-RMS	RMS	RMS	RMS	12.1
Channel-RMS	SP	SP	RMS	44.8
Expert-RMS	SP	RMS	SP	72.1
HARP	SP	SP	SP	**86.0**

(b) Allocator sensitivity

Varied	γ	η	Δ	MRP
Default	**2**	**.15**	**128**	**86.0**
γ	1/4	.15	128	85.1/81.7
η	2	0/.30	128	84.7/84.8
Δ	2	.15	64	80.8

SP is critical across all hierarchy levels. Table 2a replaces SP with RMS magnitude at different levels of the hierarchy while keeping the remaining HARP pipeline unchanged. Using SP at all three levels achieves 86.0 MRP, whereas replacing Expert-SP or Channel-SP with RMS reduces

performance to 72.1 and 44.8, respectively; using RMS throughout further drops MRP to 12.1. These results show that HARP's gains cannot be explained by heterogeneous widths or magnitude-based pruning alone. In particular, Layer-SP, Expert-SP, and Channel-SP play complementary roles: the first determines where capacity is placed across layers, the second assigns heterogeneous widths within each layer, and the third realizes each width by selecting the retained coupled channels.

The allocator is robust around the default configuration. Table 2b varies one allocation hyperparameter at a time around $(\gamma, \eta, \Delta) = (2, 0.15, 128)$. HARP remains close to its default performance when reducing γ to 1 or changing the tier-coverage floor to $\eta = 0$ or 0.30, reaching 85.1, 84.7, and 84.8 MRP, respectively. This stability indicates that the hierarchical allocator does not depend on a narrowly tuned coverage or response setting. Tier spacing has a larger effect: increasing Δ from 64 to 128 improves MRP from 80.8 to 86.0, showing that sufficient width separation allows the allocator to express more meaningful capacity differences between experts.

4.4 End-to-end heterogeneous inference

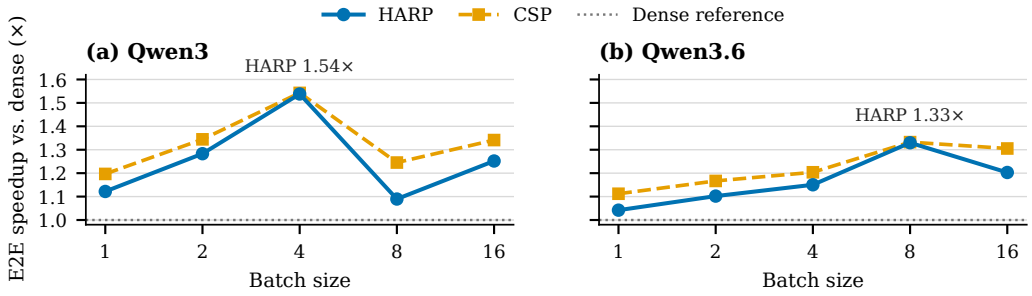

Figure 3: End-to-end inference speedup for the 50%-pruned $\Delta = 64$ Qwen3 and Qwen3.6 checkpoints on one A100-SXM4-80GB GPU with BF16 and TP=1. Each request uses a 2,048-token prompt and generates 128 tokens, with batch sizes from 1 to 16. HARP executes true heterogeneous widths through our width-aware vLLM backend, while equal-width CSP uses stock vLLM under the same protocol. Speedup is dense-model latency divided by pruned-model latency.

True-width execution translates structural compression into speedup. We implement a width-aware vLLM backend that executes each routed expert at its retained width rather than padding heterogeneous experts back to the original dimension. Figure 3 shows that this structural compression translates into measurable end-to-end acceleration on both Qwen3 and Qwen3.6 across all evaluated batch sizes. Relative to dense inference, HARP improves TTFT by 1.20–$1.37\times$ and end-to-end latency by 1.04–$1.54\times$. The speedup persists from batch 1 to 16, showing that the benefit of heterogeneous pruning is not limited to a particular serving regime. These results confirm that HARP's heterogeneous widths can be executed directly and that the reduction in routed-expert capacity yields practical inference gains rather than only nominal FLOP savings.

Heterogeneous widths retain more performance at comparable speed. The main advantage of HARP over equal-width CSP is not a larger peak speedup, but better performance retention at a similar inference cost. At the fastest measured operating point on Qwen3, both HARP and CSP reach $1.54\times$ end-to-end speedup over dense inference, while HARP retains 64.7% of dense performance compared with 59.0% for CSP. On Qwen3.6, both reach $1.33\times$ speedup, while HARP retains 80.8% versus 73.9% for CSP. Thus, the heterogeneous capacity allocation introduced by HARP converts the same practical acceleration into higher retained model quality, demonstrating that adaptive expert widths improve the accuracy–efficiency trade-off beyond uniform channel pruning.

4.5 Structural priors stabilize calibration-based channel selection

HCA improves calibration-based pruning through model-intrinsic allocation. Figure 4 evaluates Wanda and ENP across four calibration corpora under three stages: uniform-width pruning,

HCA with the same empirical channel criterion, and HCA augmented with Channel-SP. Replacing uniform widths with HCA improves MRP in 44 of 48 comparisons and reduces the four-corpus performance range in 7 of 12 model–criterion–sparsity settings. Importantly, HCA changes only where capacity is allocated across layers and experts while leaving the calibration-derived channel ranking unchanged. The consistent improvement therefore shows that model-intrinsic hierarchical allocation can strengthen calibration-based pruning without allowing the calibration corpus to determine the global capacity distribution.

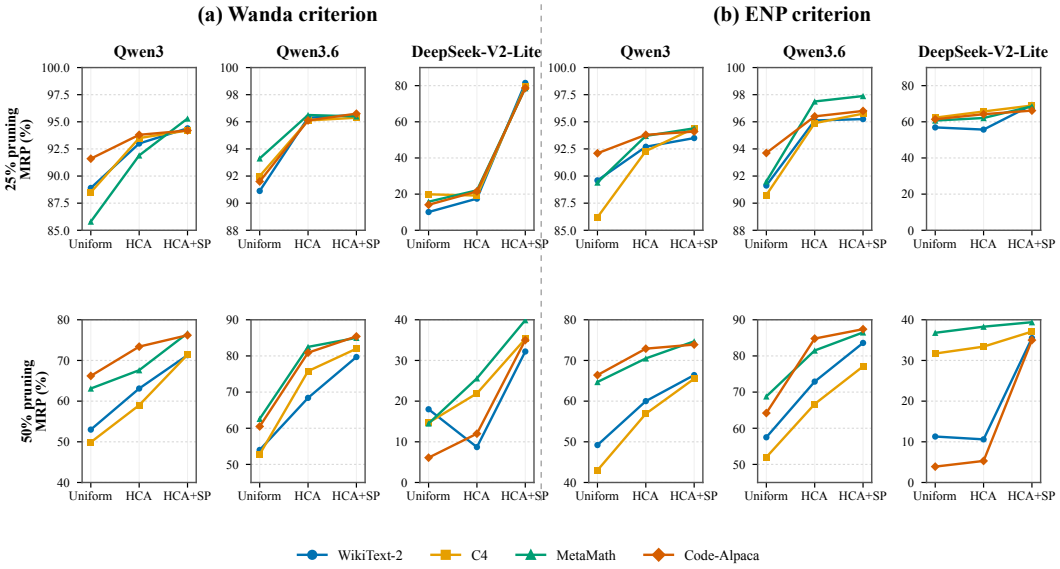

Figure 4: Absolute MRP for four calibration corpora at 25% and 50% pruning. Each Wanda or ENP panel compares Uniform, HCA allocation, and HCA+SP anchoring at fixed widths. Y-axis limits vary by model and pruning ratio but match across criteria for each setting.

Channel-SP further reduces calibration sensitivity. Adding Channel-SP to the empirical ranking under the same HCA widths further improves MRP in 47 of 48 comparisons and narrows the four-corpus range in 11 of 12 settings. The effect is particularly pronounced when the empirical criterion is unstable: on DeepSeek-V2-Lite with Wanda at 25% pruning, the four-corpus MRP range increases from only 17.5–22.1 under HCA to 78.6–81.5 after SP anchoring. These results show that the two components play complementary roles: HCA determines *where* the global capacity should be retained, while Channel-SP stabilizes *which* channels realize that capacity. Consequently, calibration evidence can refine local channel selection without controlling the overall pruning structure, making HARP substantially less sensitive to the particular calibration corpus.

5 Conclusion

Our results suggest that effective MoE channel pruning is fundamentally a capacity-placement problem rather than only a channel-ranking problem. Coordinating capacity across layers and experts becomes increasingly important as the pruning budget tightens, while model-intrinsic structural information provides a stable basis for these global decisions. Calibration signals remain useful for refining local channel selection, and true-width execution shows that the resulting heterogeneous widths can translate into practical inference gains. Our current study focuses on routed-expert channel pruning across three MoE families, with system measurements on a single A100 GPU; broader model architectures, hardware platforms, and more flexible width configurations remain useful directions for future study. An interesting direction for future work is to further explore more effective ways of utilizing and integrating calibration signals, which may complement the structural prior and further improve local channel selection. Overall, these findings support hierarchical, structure-aware capacity redistribution as a practical approach to MoE compression.

Reproducibility Statement

We provide the theoretical, algorithmic, experimental, and system details needed to reproduce the reported results. The derivation and mathematical properties of the structural preservation score are given in Appendix A, while Appendix B specifies the parameter groups and discrete allocation procedure. Model identifiers, pruning configurations, datasets, calibration protocols, and evaluation settings are documented in Appendix C. Appendix D reports the complete ablation results, and Appendix E describes the true-width inference backend and provides the raw latency measurements underlying the efficiency results. Additional structural analyses are reported in Appendix F. The complete implementation will be released publicly upon acceptance.

AI Use Statement

Generative AI tools, including OpenAI ChatGPT and Codex, were used to assist with manuscript organization, English drafting and editing, literature retrieval, preparation of the LaTeX source, and refinement of mathematical exposition. AI tools were also used to help complete intermediate derivation steps and check algebraic consistency. The research ideas, methodology, mathematical derivations, experimental design, and interpretation of experimental results were developed by the authors. All AI-assisted text, mathematical content, literature references, and technical descriptions were reviewed and verified by the authors, who take full responsibility for the correctness and final content of the paper.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Abhinav Bandari, Lu Yin, Cheng-Yu Hsieh, Ajay Kumar Jaiswal, Tianlong Chen, Li Shen, Ranjay Krishna, and Shiwei Liu. Is C4 dataset optimal for pruning? an investigation of calibration data for LLM pruning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18089–18099, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1004. URL https://aclanthology.org/2024.emnlp-main.1004/.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, 2021. URL https://arxiv.org/abs/2107.03374.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao

Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model, 2024. URL https://arxiv.org/abs/2405.04434.

Yifu Ding, Jiacheng Wang, Ge Yang, Yongcheng Jing, Jinyang Guo, Xianglong Liu, and Dacheng Tao. Attribution-Guided and Coverage-Maximized Pruning for Structural MoE Compression, 2026. URL https://arxiv.org/abs/2606.18304.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Jiangyang He, Shaolin Zhu, and Deyi Xiong. TENP: Trapezoidal expert neuron pruning for mixture-of-experts. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20911–20925, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1049. URL https://aclanthology.org/2026.findings-acl.1049/.

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*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://dl.acm.org/doi/10.1145/3600006.3613165.

Zongfang Liu, Guangyi Chen, Shengkun Tang, Yifan Shen, Huan Wang, and Xin Yuan. AIMER: Calibration-free task-agnostic MoE expert pruning, 2026a. URL https://arxiv.org/abs/2603.18492.

Zongfang Liu, Shengkun Tang, Boyang Sun, Zhiqiang Shen, and Xin Yuan. EvoESAP: Non-Uniform Expert Pruning for Sparse MoE, 2026b. 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.

ModelScope Team. EvalScope: Evaluation framework for large models, 2024. URL https://github.com/modelscope/evalscope.

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. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. EvoPress: Accurate Dynamic Model Compression via Evolutionary Search. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55556–55590. PMLR, 2025. URL https://proceedings.mlr.press/v267/sieberling25a.html.

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*, pp. 4942–4964, 2024a. URL https://openreview.net/forum?id=PxoFut3dWW.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=PxoFut3dWW.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning Mixture-of-Experts Large Language Model using the Hints from Its Router, 2024. URL https://arxiv.org/abs/2410.12013.

Hu Xu, Zhaolong Xing, Congcong Liu, Jiaxing Wang, Zhida Jiang, Junshi Huang, Zhen Chen, and Jianfeng Xu. Averaged evaluation masks capability trade-offs: Multi-source calibration for high-sparsity LLM pruning, 2026. URL https://arxiv.org/abs/2606.03328.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57101–57115. PMLR, 2024. URL https://proceedings.mlr.press/v235/yin24e.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

A Score Derivation and Properties

A.1 Preservation risk under independent thinning

Throughout the derivation, let $\Theta = (\theta_i)_{i=1}^N \neq 0$. For a retention probability $q \in (0,1)$, let m_i be independent Bernoulli random variables, $m_i \sim \mathrm{Bernoulli}(q)$, and define the retained absolute-weight mass

$$Z_q = \sum_{i=1}^{N} m_i |\theta_i|. \tag{13}$$

By linearity of expectation and independence,

$$\mathbb{E}[Z_q] = q \sum_i |\theta_i| = q\|\Theta\|_1, \tag{14}$$

$$\mathrm{Var}(Z_q) = \sum_i \theta_i^2 \,\mathrm{Var}(m_i) = q(1-q)\|\Theta\|_2^2. \tag{15}$$

Therefore, the squared coefficient of variation of the retained mass is

$$\mathrm{CV}^2(Z_q) = \frac{\mathrm{Var}(Z_q)}{\mathbb{E}[Z_q]^2} = \frac{1-q}{q}\frac{\|\Theta\|_2^2}{\|\Theta\|_1^2}. \tag{16}$$

For an equal-magnitude parameter group of the same size N, the corresponding relative variance is $(1-q)/(qN)$. Normalizing by this reference removes both the retention probability and the group dimension, yielding the structural preservation score

$$\mathrm{SP}(\Theta) = \frac{qN}{1-q}\,\mathrm{CV}^2(Z_q) = \frac{N\|\Theta\|_2^2}{\|\Theta\|_1^2}, \tag{17}$$

which is exactly the score used in Equation (5). In particular, $\mathrm{SP}(\Theta)$ is independent of the thinning probability q.

The random-thinning construction provides a statistical interpretation of the weight statistic: a larger $\mathrm{SP}(\Theta)$ corresponds to larger relative fluctuations of retained absolute-weight mass under random coordinate removal, and hence a higher preservation priority. HARP itself does not sample random masks during pruning; layer allocation, expert-width assignment, and coupled-channel selection are deterministic once the pretrained weights and allocator hyperparameters are fixed.

A.2 Equivalent weight-dispersion identity

Let X be the random variable that takes value $|\theta_i|$ with probability $1/N$. Its first two moments are

$$\mathbb{E}[X] = \frac{1}{N}\sum_i |\theta_i|, \qquad \mathbb{E}[X^2] = \frac{1}{N}\sum_i \theta_i^2. \tag{18}$$

Hence

$$1 + \mathrm{CV}^2(X) = \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2} \tag{19}$$

$$= \frac{N\sum_i \theta_i^2}{\left(\sum_i |\theta_i|\right)^2} \tag{20}$$

$$= \mathrm{SP}(\Theta). \tag{21}$$

Thus, the same score can be interpreted equivalently as one plus the squared coefficient of variation of coordinate magnitudes. This identity characterizes the concentration of absolute parameter mass, whereas the independent-thinning construction above supplies the preservation-risk interpretation used by HARP.

A.3 SP PROPERTIES

The score is invariant to global rescaling. For any $\alpha \neq 0$,

$$\mathrm{SP}(\alpha\Theta) = \frac{N\|\alpha\Theta\|_2^2}{\|\alpha\Theta\|_1^2} = \frac{N|\alpha|^2\|\Theta\|_2^2}{|\alpha|^2\|\Theta\|_1^2} = \mathrm{SP}(\Theta). \tag{22}$$

The score is also bounded. By Cauchy–Schwarz,

$$\|\Theta\|_1^2 \leq N\|\Theta\|_2^2, \tag{23}$$

which implies $\mathrm{SP}(\Theta) \geq 1$. Equality holds exactly when all coordinate magnitudes are equal. Conversely, since

$$\left(\sum_i |\theta_i|\right)^2 = \sum_i \theta_i^2 + 2\sum_{i<j} |\theta_i||\theta_j| \geq \sum_i \theta_i^2, \tag{24}$$

we have $\|\Theta\|_1^2 \geq \|\Theta\|_2^2$, and therefore $\mathrm{SP}(\Theta) \leq N$. Equality holds when at most one coordinate is nonzero. Thus,

$$1 \leq \mathrm{SP}(\Theta) \leq N. \tag{25}$$

The score also admits an effective-support interpretation. Define normalized absolute masses

$$p_i = \frac{|\theta_i|}{\|\Theta\|_1}, \qquad \sum_i p_i = 1, \tag{26}$$

and the effective number of participating coordinates

$$N_{\mathrm{eff}} = \frac{1}{\sum_i p_i^2}. \tag{27}$$

Then

$$\mathrm{SP}(\Theta) = N\sum_i p_i^2 = \frac{N}{N_{\mathrm{eff}}}. \tag{28}$$

Thus, SP measures the ratio between the nominal group size and its effective support. A uniformly distributed group has $N_{\mathrm{eff}} = N$ and $\mathrm{SP} = 1$, whereas a group whose absolute mass is concentrated on a single coordinate has $N_{\mathrm{eff}} = 1$ and $\mathrm{SP} = N$.

For example, $\Theta = (1, 1, 1, 1)$ gives $N_{\mathrm{eff}} = 4$ and $\mathrm{SP} = 1$, while $\Theta = (4, 0, 0, 0)$ gives $N_{\mathrm{eff}} = 1$ and $\mathrm{SP} = 4$. Replicating every coordinate the same number of times leaves SP unchanged, because both the nominal group size N and the effective support N_{eff} scale by the same factor.

A.4 PARENT SCORES AND CHILD STRUCTURE

Partition a parent parameter group into disjoint child groups $\{\Theta_j\}_j$. Let

$$a_j = \|\Theta_j\|_1, \qquad b_j = \|\Theta_j\|_2^2, \qquad N_j = |\Theta_j|, \qquad S_j = \mathrm{SP}(\Theta_j). \tag{29}$$

By the definition of SP,

$$b_j = \frac{a_j^2 S_j}{N_j}. \tag{30}$$

Let

$$N_{\mathrm{parent}} = \sum_j N_j. \tag{31}$$

Since the child groups are disjoint,

$$\|\Theta_{\mathrm{parent}}\|_1 = \sum_j a_j, \qquad \|\Theta_{\mathrm{parent}}\|_2^2 = \sum_j b_j. \tag{32}$$

The parent score therefore satisfies

$$S_{\text{parent}} = \frac{N_{\text{parent}} \sum_j b_j}{\left(\sum_j a_j\right)^2} \tag{33}$$

$$= N_{\text{parent}} \sum_j \left(\frac{a_j}{\sum_k a_k}\right)^2 \frac{S_j}{N_j}. \tag{34}$$

Equation (34) shows that a parent score is not an arithmetic mean of its child scores. It depends jointly on the child SP values, their absolute-weight masses, and their dimensions. Consequently, Layer-SP and Expert-SP cannot in general be recovered by simply averaging Expert-SP and Channel-SP values, respectively. This motivates computing the same SP functional independently at each structural granularity, as done in Section 3.

The identity also explains why removing the highest-scoring child does not necessarily decrease the recomputed parent score. Such a removal changes the parent's squared ℓ_2 mass, absolute ℓ_1 mass, and parameter count simultaneously. Accordingly, the structural analyses that remove experts or channels recompute SP from the remaining parameter moments rather than manipulating a list of child scores.

A.5 Monotone capacity matching

The structural preservation score supplies an ordering over structural units, while the global budget determines the available capacity values. The following exchange argument formalizes the monotone assignment used by HARP once a feasible multiset of widths has been determined.

Theorem 1 (Monotone capacity matching). *Let structure i have preservation priority $a_i \geq 0$, and let its width-dependent surrogate loss be $a_i g(K_i)$, where g is nonincreasing in the assigned width K_i. For any fixed multiset of widths, there exists an optimal assignment that matches larger widths to larger preservation priorities. In particular, if $a_i \geq a_j$, an ordered optimal assignment satisfies $K_i \geq K_j$.*

Proof. Consider $a_i \geq a_j$ and two widths $K_h \geq K_l$. Assigning the larger width to i rather than j changes their combined surrogate loss by

$$\big[a_i g(K_h) + a_j g(K_l)\big] - \big[a_i g(K_l) + a_j g(K_h)\big] \tag{35}$$

$$= (a_i - a_j)\big[g(K_h) - g(K_l)\big] \leq 0, \tag{36}$$

because $a_i - a_j \geq 0$ and $g(K_h) - g(K_l) \leq 0$. Therefore, swapping any inverted pair cannot increase the objective. Repeating this exchange eliminates all inversions and yields an ordered optimal assignment. □

Taking

$$a_i = \text{SP}(\Theta_i) \tag{37}$$

makes the preservation-priority order exactly the SP order. Theorem 1 therefore implies

$$S_i^E > S_j^E \quad \Longrightarrow \quad K_i \geq K_j, \tag{38}$$

which is the Expert-SP tier assignment used after the budget-constrained search determines the numbers of low-, middle-, and high-width experts.

At the layer level, HARP applies the monotone transformation $(S_\ell^L)^\gamma$ before normalized budget redistribution. Since $\gamma > 0$, this transformation preserves the Layer-SP ordering. Water filling, clipping, tier realization, and remainder propagation determine the feasible capacity magnitudes while preserving the global budget.

Proposition 1 (Optimal SP-guided channel selection). *For channel priorities u_c and a fixed retained width K, the set containing the K largest priorities maximizes*

$$\sum_{c \in \mathcal{C}} u_c \tag{39}$$

over all subsets $\mathcal{C}$ satisfying $|\mathcal{C}| = K$.

Proof. Suppose a feasible set contains channel j but excludes channel i with $u_i > u_j$. Replacing j by i strictly increases the objective. Hence, no maximizing set can contain such an inversion, and every maximizer is a top-K set up to ties. □

With

$$u_c = S^C_{\ell,e,c}, \tag{40}$$

Proposition 1 shows that Channel-SP top-K selection exactly maximizes the additive retained-priority surrogate at the realized expert width $K_{\ell,e}$. Together, Theorem 1 and Proposition 1 clarify how the common SP functional is used across the hierarchy: the budget mechanism determines feasible capacity quantities, monotone matching assigns those capacities to higher-priority structures, and top-K selection realizes each expert width at the channel level.

B Allocation and Implementation Details

This section specifies the discrete realization used for the calibration-free HARP results. Section 3 introduces the structural priorities and allocation principle.

Width tiers. For pruning ratio s, the candidate widths are

$$K_{\text{mid}} = D(1-s), \qquad K_{\text{low}} = K_{\text{mid}} - \Delta, \qquad K_{\text{high}} = K_{\text{mid}} + \Delta, \tag{41}$$

where D is the original intermediate width. The main experiments use $(\gamma, \eta, \Delta) = (2, 0.15, 128)$, with widths restricted to valid hardware-aligned values when required.

Structural priority score. For a parameter group Θ with N scalar parameters, HARP uses

$$\rho(\Theta) = \frac{N\|\Theta\|_2^2}{\|\Theta\|_1^2}. \tag{42}$$

and defines $\rho(\Theta) = 0$ when $\|\Theta\|_1 = 0$. Applying this functional to the layer, expert, and coupled-channel groups in Equation (6) gives ρ^L_ℓ, $\rho^E_{\ell,e}$, and $\rho^C_{\ell,e,c}$. Algorithm 1 uses the layer and expert scores to realize discrete widths.

The layerwise realization can leave a small global residual because each expert is restricted to one of three discrete width tiers. Algorithm 2 closes this residual exactly and then realizes the retained channel sets using Channel-SP.

Layer targets and expert tiers. ProjectLayerTargets clips continuous layer targets to $[K_{\text{low}}, K_{\text{high}}]$ while enforcing

$$\sum_{\ell=1}^{L} K_\ell^{\text{target}} = LK_{\text{mid}}. \tag{43}$$

Any surplus or deficit is redistributed by Layer-SP priority before expert-level discretization. For each layer, FeasibleTierCounts enumerates integer triples (n_h, n_m, n_l) satisfying

$$n_h + n_m + n_l = E, \tag{44}$$

and

$$n_h K_{\text{high}} + n_m K_{\text{mid}} + n_l K_{\text{low}} \le B_\ell. \tag{45}$$

Whenever feasible, we additionally impose the coverage constraint

$$n_h, n_m, n_l \ge \lceil \eta E \rceil, \tag{46}$$

If no feasible triple satisfies the coverage floor, the floor is removed. Among feasible candidates, SelectTierCounts first maximizes

$$n_h K_{\text{high}} + n_m K_{\text{mid}} + n_l K_{\text{low}}, \tag{47}$$

then favors a more balanced three-tier population. Expert-SP assigns wider tiers to higher-priority experts.

Require: Layer scores $\{\rho_\ell^L\}_{\ell=1}^L$, expert scores $\{\rho_{\ell,e}^E\}$, pruning ratio s, γ, η, Δ
Ensure: Discrete expert widths $\{K_{\ell,e}\}$
$K_{\text{mid}} \leftarrow D(1-s)$
$K_{\text{low}} \leftarrow K_{\text{mid}} - \Delta$
$K_{\text{high}} \leftarrow K_{\text{mid}} + \Delta$
Continuous layer budgets
for $\ell = 1$ to L**:**
$\alpha_\ell \leftarrow \dfrac{L(\rho_\ell^L)^\gamma}{\sum_{j=1}^L (\rho_j^L)^\gamma}$
$\widetilde{K}_\ell \leftarrow K_{\text{low}} + \alpha_\ell(K_{\text{mid}} - K_{\text{low}})$

$\{K_\ell^{\text{target}}\}_{\ell=1}^L \leftarrow \textsc{ProjectLayerTargets}(\{\widetilde{K}_\ell\}, K_{\text{low}}, K_{\text{high}}, LK_{\text{mid}}, \{\rho_\ell^L\})$
Discrete expert-tier realization
$\pi \leftarrow \textsc{StableSortDesc}(\{\rho_\ell^L\}_{\ell=1}^L)$
$R \leftarrow 0$
for ℓ in π**:**
$B_\ell \leftarrow EK_\ell^{\text{target}} + R$
$\mathcal{F} \leftarrow \textsc{FeasibleTierCounts}(B_\ell, E, K_{\text{low}}, K_{\text{mid}}, K_{\text{high}}, \lceil \eta E \rceil)$
if $\mathcal{F} = \varnothing$**:**
$\mathcal{F} \leftarrow \textsc{FeasibleTierCounts}(B_\ell, E, K_{\text{low}}, K_{\text{mid}}, K_{\text{high}}, 0)$

$(n_h, n_m, n_l) \leftarrow \textsc{SelectTierCounts}(\mathcal{F})$
$\sigma_\ell \leftarrow \textsc{StableSortDesc}(\{\rho_{\ell,e}^E\}_{e=1}^E)$
for $r = 1$ to E**:**
if $r \le n_h$**:**
$K_{\ell,\sigma_\ell(r)} \leftarrow K_{\text{high}}$
else if $r \le n_h + n_m$**:**
$K_{\ell,\sigma_\ell(r)} \leftarrow K_{\text{mid}}$
else:
$K_{\ell,\sigma_\ell(r)} \leftarrow K_{\text{low}}$

$R \leftarrow B_\ell - \sum_{e=1}^E K_{\ell,e}$

return $\{K_{\ell,e}\}$

Algorithm 1: HARP hierarchical width allocation

Residual closure and channel realization. The residual

$$R = B_\ell - \sum_{e=1}^{E} K_{\ell,e} \tag{48}$$

is propagated in descending Layer-SP order. After all layers, the global deficit is

$$M = LEK_{\text{mid}} - \sum_{\ell=1}^{L}\sum_{e=1}^{E} K_{\ell,e}. \tag{49}$$

Equal tier spacing makes M a nonnegative integer multiple of Δ, so HARP promotes exactly

$$U = \frac{M}{\Delta} \tag{50}$$

eligible experts by one tier, following Layer-SP priority and then Expert-SP priority. This yields

$$\sum_{\ell=1}^{L}\sum_{e=1}^{E} K_{\ell,e} = LEK_{\text{mid}} \tag{51}$$

Require: Expert widths $\{K_{\ell,e}\}$, layer scores $\{\rho^L_\ell\}$, expert scores $\{\rho^E_{\ell,e}\}$, channel scores $\{\rho^C_{\ell,e,c}\}$, K_{mid}, K_{high}, Δ
Ensure: Budget-closed expert widths $\{K_{\ell,e}\}$ and retained channel sets $\{\mathcal{C}_{\ell,e}\}$

Exact global-budget closure

$M \leftarrow LEK_{\text{mid}} - \sum_{\ell=1}^{L}\sum_{e=1}^{E} K_{\ell,e}$
$U \leftarrow M/\Delta$
$\mathcal{P} \leftarrow \{(\ell,e) : K_{\ell,e} < K_{\text{high}}\}$
$\mathcal{P} \leftarrow \textsc{StableSort}(\mathcal{P}; (\text{rank}_L(\ell), \text{rank}_E(\ell,e), \ell, e))$
for $i = 1$ to U**:**
$(\ell, e) \leftarrow \mathcal{P}_i$
$K_{\ell,e} \leftarrow K_{\ell,e} + \Delta$

Width-conditioned channel realization

for $\ell = 1$ to L**:**
for $e = 1$ to E**:**
$\sigma^C_{\ell,e} \leftarrow \textsc{StableSortDesc}(\{\rho^C_{\ell,e,c}\}_{c=1}^{D})$
$\mathcal{C}_{\ell,e} \leftarrow \{\sigma^C_{\ell,e}(1), \ldots, \sigma^C_{\ell,e}(K_{\ell,e})\}$

return $\{K_{\ell,e}\}, \{\mathcal{C}_{\ell,e}\}$

Algorithm 2: Global budget closure and channel realization

without introducing widths outside the three-tier set. At each realized width, Channel-SP defines

$$\mathcal{C}_{\ell,e} = \text{TopK}\left(\{\rho^C_{\ell,e,c}\}_{c=1}^{D}, K_{\ell,e}\right). \tag{52}$$

The retained gate and up rows and down-projection columns share these indices, preserving the coupled channel structure.

Deterministic ranking. All rankings use stable sorting. Original expert and channel indices resolve exact Expert-SP and Channel-SP ties, making realization deterministic.

Calibration-assisted variants. Calibration-assisted variants reuse exactly the same layer targets, expert-tier assignments, residual closure, and final widths $\{K_{\ell,e}\}$ produced by Algorithms 1 and 2. They modify only the channel ordering in the realization step, replacing Channel-SP with the anchored score in Equation (12). Thus, calibration data do not affect layer budgets, expert widths, or the global channel budget.

C Models and Evaluation Protocol

C.1 Models and pruning configurations

Table 3: Evaluated checkpoints and routed-expert layouts. D denotes the original routed-expert intermediate width. DeepSeek layer 0 is dense and excluded from pruning.

Checkpoint	Projection layout	MoE layers	Experts/layer	D
`Qwen3-30B-A3B-Instruct-2507`	separate	48	128	768
`Qwen3.6-35B-A3B`	packed gate–up	40	256	512
`DeepSeek-V2-Lite-Chat`	separate	26	64	1408

We evaluate 25% and 50% routed-channel pruning for all three model families. Only routed-expert intermediate channels are pruned. Removing channel c jointly slices $W^g[c,:]$, $W^u[c,:]$, and $W^d[:,c]$. For Qwen3.6, whose gate and up projections are packed, the same channel index is applied to the corresponding semantic gate and up halves before repacking. Attention modules, routers, shared experts, non-MoE layers, and all other model components remain unchanged.

Table 4: Aligned routed-expert channel widths under the two tier spacings. Uniform-width methods use K_0. Both HSP and HARP use $\Delta = 128$ in the main comparison, while $\Delta = 64$ is used only in the ablation and efficiency studies.

Family	Pruning	Uniform K_0	$\Delta = 64$ tiers	$\Delta = 128$ tiers
Qwen3	25%	576	512/576/640	448/576/704
Qwen3	50%	384	320/384/448	256/384/512
Qwen3.6	25%	384	320/384/448	256/384/512
Qwen3.6	50%	256	192/256/320	128/256/384
DeepSeek	25%	1056	992/1056/1120	928/1056/1184
DeepSeek	50%	704	640/704/768	576/704/832

AIMER removes complete routed experts rather than pruning expert channels. At 25%/50% pruning, it retains 96/64 of 128 experts per Qwen3 layer, 192/128 of 256 experts per Qwen3.6 layer, and 48/32 of 64 experts per DeepSeek MoE layer.

C.2 Task and execution configuration

Table 5: Evaluation settings for the eight-task suite.

Field	Specification
Evaluation path	Hugging Face checkpoint → vLLM → EvalScope `openai_api`
Sampling	seed 42; no shuffle; temperature 0; greedy decoding
Chat template	thinking explicitly disabled for every request
Batch and timeout	evaluation batch size 16; timeout 1200 seconds
Coverage	complete splits; 31,382 examples; no sample limit
Serving	CUDA 12.8; maximum model length 8192; GPU memory utilization 0.90
DeepSeek adapter	trust remote code; separate routed and shared intermediate-size fields

Table 6: Datasets in the eight-task evaluation suite, in evaluation order. Generation limits apply per example and do not reduce the number of evaluated examples.

Dataset	Examples	Max tokens	Setting
ARC	3548	2048	Challenge + Easy
HellaSwag	10042	512	full validation split
WinoGrande	1267	1024	full validation split
GSM8K	1319	2048	0-shot
MATH-500	500	4096	full test split
MMLU	14042	2048	all subjects
HumanEval	164	1024	pass@1
MBPP	500	1024	3-shot, pass@1

Calibration and baseline implementation. Wanda and ENP are evaluated under four matched calibration protocols using WikiText-2, C4, MetaMath, and Code-Alpaca. Each protocol uses 128 sequences of length 2048. Calibration caches are constructed independently for each model with its native tokenizer under the same cache-construction procedure, while varying only the source corpus.

The main-table Wanda and ENP results average over the four calibration corpora. Their task-level results in Tables 7–9 use the same convention, averaging each task across the four calibration settings before computing the retained-performance aggregate. For standalone Wanda and ENP, experts receiving no calibration tokens use a deterministic coupled weight-ℓ_2 fallback. Random uses seed 42 independently within each expert. AIMER, CSP, HSP, and HARP require no calibration data. All rankings are computed before downstream evaluation and remain fixed across the eight tasks.

Wanda applies its structured channel score independently within each expert and retains equal top-K widths. ENP applies the neuron score from TENP under the same equal-width rule; both methods reuse one channel ordering across the two pruning ratios. AIMER ranks complete routed experts by its calibration-free weight statistic and removes the same number from each MoE layer. CSP retains the top Channel-SP prefix at the uniform width K_0. HSP assigns the high, middle, and low tiers to the top 25%, middle 50%, and bottom 25% of experts by Expert-SP within every layer. HARP uses the same Expert-SP and Channel-SP priorities, but determines layer budgets and per-layer tier counts through Layer-SP water filling, budget-constrained tier search, and remainder propagation.

C.3 TASK-LEVEL DOWNSTREAM RESULTS

Tables 7–9 report the task-level results corresponding to the main comparison in Table 1. For Wanda and ENP, each task score is averaged over the four matched calibration corpora: WikiText-2, C4, MetaMath, and Code-Alpaca. Random, AIMER, CSP, HSP, and HARP do not require calibration data. ARC, HellaSwag, WinoGrande, GSM8K, MATH-500, and MMLU are reported as accuracy (%), while HumanEval and MBPP are reported as pass@1 (%). "Ret." denotes the unweighted mean percentage retained relative to the dense checkpoint, consistent with the definition used in Table 1. AIMER is included as a cross-granularity whole-expert reference.

Table 7: Qwen3-30B-A3B-Instruct-2507 results on the eight-task evaluation suite.

Pruning	Method	ARC Acc.	Hella Acc.	Wino Acc.	GSM8K Acc.	MATH Acc.	MMLU Acc.	HEval Pass@1	MBPP Pass@1	Ret. %
0%	Dense	97.35	87.09	75.37	94.92	89.20	85.52	95.12	78.80	100.0
25%	Random	95.24	81.07	67.48	83.78	68.60	78.03	75.61	67.60	87.8
	Wanda	95.26	83.01	67.80	89.61	78.10	78.60	67.68	64.25	88.7
	AIMER	95.13	84.00	72.22	93.63	86.00	74.15	91.46	75.20	95.4
	ENP	95.16	83.46	67.84	91.00	76.35	77.56	71.95	65.45	89.3
	CSP	95.83	82.74	68.19	93.40	79.20	79.32	80.49	68.20	91.9
	HSP	95.78	83.15	68.82	94.16	81.60	80.15	82.93	69.20	93.1
	HARP	95.64	83.04	70.32	94.39	86.00	79.66	88.41	70.40	**94.8**
50%	Random	85.29	56.49	54.93	34.27	15.20	44.38	16.46	16.80	46.1
	Wanda	82.02	64.21	55.23	63.68	33.90	57.56	24.09	27.80	58.1
	AIMER	82.13	72.34	62.98	71.57	53.80	52.11	60.37	49.80	71.8
	ENP	79.33	61.02	55.53	58.32	27.65	50.40	29.57	30.55	55.8
	CSP	80.55	53.75	51.62	73.16	33.20	57.67	32.93	33.40	59.0
	HSP	81.46	55.56	53.04	76.95	38.80	57.52	39.02	34.80	61.9
	HARP	85.43	59.58	56.51	81.88	51.00	60.99	51.83	42.80	**69.3**

Table 8: Qwen3.6-35B-A3B results on the eight-task evaluation suite.

Pruning	Method	ARC Acc.	Hella Acc.	Wino Acc.	GSM8K Acc.	MATH Acc.	MMLU Acc.	HEval Pass@1	MBPP Pass@1	Ret. %
0%	Dense	98.31	89.86	87.13	96.51	94.80	89.09	95.73	80.00	100.0
25%	Random	97.27	85.52	82.79	90.98	83.40	82.82	82.93	68.20	92.0
	Wanda	96.27	89.18	77.58	86.81	81.15	82.78	89.17	70.35	92.0
	AIMER	97.98	89.95	83.50	95.98	90.80	86.30	94.51	75.40	97.6
	ENP	96.87	89.56	77.40	93.22	84.90	83.50	82.01	65.35	91.8
	CSP	97.15	88.88	77.74	95.07	89.80	85.26	85.37	69.40	94.0
	HSP	97.77	89.29	79.01	96.06	90.40	85.74	90.85	73.40	95.9
	HARP	98.05	89.21	80.11	95.53	93.00	84.59	93.29	76.80	**97.1**
50%	Random	92.27	78.41	65.35	49.20	31.60	55.36	32.32	19.20	57.5
	Wanda	89.35	71.42	67.36	28.24	26.50	55.81	44.21	37.00	57.5
	AIMER	93.01	68.47	79.64	44.73	60.80	62.04	60.37	59.00	72.4
	ENP	91.24	80.04	68.82	67.51	36.20	57.52	13.11	30.15	60.6
	CSP	93.68	79.65	67.80	82.56	53.80	71.22	48.17	45.40	73.9
	HSP	94.81	82.97	71.82	91.58	68.20	74.38	59.15	49.00	80.6
	HARP	95.57	83.39	74.27	93.48	77.40	74.87	73.17	59.20	**86.0**

Table 9: DeepSeek-V2-Lite-Chat results on the eight-task evaluation suite.

Pruning	Method	ARC Acc.	Hella Acc.	Wino Acc.	GSM8K Acc.	MATH Acc.	MMLU Acc.	HEval Pass@1	MBPP Pass@1	Ret. %
0%	Dense	79.25	66.22	55.49	70.58	28.80	56.13	51.83	46.80	100.0
25%	Random	68.04	51.81	53.28	32.68	6.80	47.64	14.63	23.60	61.7
	Wanda	13.04	12.74	15.25	2.58	1.15	24.55	0.00	2.40	15.0
	AIMER	64.38	47.52	51.93	50.27	12.20	42.95	18.29	18.60	64.0
	ENP	64.50	44.94	52.80	34.17	7.20	46.00	18.90	21.80	60.3
	CSP	69.48	50.87	53.67	45.72	11.00	44.84	17.68	22.60	65.8
	HSP	70.49	55.84	54.78	44.73	11.60	45.79	19.51	24.00	68.3
	HARP	70.55	52.19	55.17	49.20	13.80	46.18	21.34	22.20	**69.5**
50%	Random	0.00	0.00	0.00	0.08	0.00	0.88	0.00	0.00	0.2
	Wanda	15.12	13.39	15.15	0.68	0.80	19.71	0.00	0.60	13.3
	AIMER	20.41	20.66	41.67	3.56	1.00	26.61	0.00	0.60	23.7
	ENP	25.18	20.10	26.30	1.84	1.40	26.05	0.30	1.65	20.9
	CSP	44.78	29.00	51.78	4.70	2.40	35.77	5.49	7.00	37.2
	HSP	46.99	32.27	51.07	2.20	2.80	35.31	1.83	7.80	37.0
	HARP	46.14	30.78	51.70	6.67	3.40	34.46	4.88	7.20	**38.2**

D Ablation Details

All ablations use Qwen3.6-35B-A3B at 50% routed-channel pruning unless a table names another setting. They retain the eight-task evaluation protocol, global channel budget, alignment, coupled gate/up/down slicing, and deterministic ranking used in the main comparison.

D.1 Structural scoring decisions

Table 10 expands the structural-score ablation reported in Table 2. All variants use Qwen3.6-35B-A3B at 50% routed-channel pruning and keep the same global channel budget, three-tier width set, hierarchical allocator, coupled gate/up/down slicing, and downstream evaluation protocol. Only the structural scoring function at the indicated hierarchy level is changed.

For a parameter group $\Theta \in \mathbb{R}^N$, the default HARP configuration uses

$$\mathrm{SP}(\Theta) = \frac{N\|\Theta\|_2^2}{\|\Theta\|_1^2},$$

while the RMS replacement uses

$$\mathrm{RMS}(\Theta) = \frac{\|\Theta\|_2}{\sqrt{N}}.$$

Expert-RMS replaces Expert-SP while retaining Layer-SP and Channel-SP; Channel-RMS replaces Channel-SP while retaining Layer-SP and Expert-SP; and All-RMS uses RMS at all three hierarchy levels. The default HARP configuration uses SP for layer allocation, expert-width assignment, and within-expert channel selection.

Table 10: Full task results for the structural-score ablation on Qwen3.6-35B-A3B at 50% routed-channel pruning. Task scores are fractions. "Ret." denotes the unweighted mean percentage retained relative to the dense checkpoint.

Variant	Layer	Expert	Channel	ARC	Hella	Wino	GSM8K	MATH	MMLU	HEval	MBPP	Ret.
All-RMS	RMS	RMS	RMS	.0595	.0013	.0008	.3343	.3500	.0217	.1280	.0220	12.1
Channel-RMS	SP	SP	RMS	.3450	.3109	.0489	.7233	.5400	.5409	.2683	.5000	44.8
Expert-RMS	SP	RMS	SP	.9591	.8214	.7135	.7422	.7520	.6121	.7073	.0560	72.1
HARP	SP	SP	SP	.9557	.8339	.7427	.9348	.7740	.7487	.7317	.5920	**86.0**

Using SP at all three hierarchy levels achieves 86.0% retained performance. Replacing Expert-SP with RMS reduces the aggregate to 72.1%, while replacing Channel-SP reduces it further to 44.8%. Using RMS throughout the hierarchy yields only 12.1%. These controlled replacements keep the allocation pipeline otherwise unchanged, showing that the observed gains are not explained by heterogeneous expert widths alone. Instead, the layer, expert, and channel instances of SP play

complementary roles in determining where capacity is allocated, which experts receive wider tiers, and which coupled channels are retained within each realized width.

D.2 TIER SPACING ACROSS MODELS

We next examine the effect of the tier spacing Δ across all three model families and both pruning ratios. We compare $\Delta = 64$ and $\Delta = 128$ while keeping the remaining HARP configuration fixed. For a given pruning ratio, the middle width K_{mid} is determined by the target global channel budget, and the low- and high-width tiers are

$$K_{\mathrm{low}} = K_{\mathrm{mid}} - \Delta, \qquad K_{\mathrm{high}} = K_{\mathrm{mid}} + \Delta.$$

Thus, increasing Δ enlarges the capacity difference between the low-, middle-, and high-width experts while preserving the same global routed-channel budget.

Table 11: Task-level comparison of tier spacings $\Delta = 64$ and $\Delta = 128$. Task scores are fractions. "Ret." denotes the unweighted mean percentage retained relative to the corresponding dense checkpoint.

Pruning	Δ	ARC	Hella	Wino	GSM8K	MATH	MMLU	HEval	MBPP	Ret.
Qwen3-30B-A3B-Instruct-2507										
25%	64	.9594	.8322	.6914	.9378	.8180	.8029	.8476	.6720	93.1
25%	128	.9564	.8304	.7032	.9439	.8600	.7966	.8841	.7040	**94.8**
50%	64	.8261	.5645	.5478	.7854	.4440	.5753	.4390	.3860	64.7
50%	128	.8543	.5958	.5651	.8188	.5100	.6099	.5183	.4280	**69.3**
Qwen3.6-35B-A3B										
25%	64	.9758	.8912	.7901	.9553	.9140	.8554	.9024	.7480	96.0
25%	128	.9805	.8921	.8011	.9553	.9300	.8459	.9329	.7680	**97.1**
50%	64	.9453	.8231	.7056	.9121	.6840	.7440	.6280	.4980	80.8
50%	128	.9557	.8339	.7427	.9348	.7740	.7487	.7317	.5920	**86.0**
DeepSeek-V2-Lite-Chat										
25%	64	.6978	.5110	.5556	.4511	.1200	.4634	.1890	.2340	67.5
25%	128	.7055	.5219	.5517	.4920	.1380	.4618	.2134	.2220	**69.5**
50%	64	.4752	.3092	.5320	.0318	.0240	.3587	.0366	.0660	37.6
50%	128	.4614	.3078	.5170	.0667	.0340	.3446	.0488	.0720	**38.2**

Across the six model–pruning configurations, $\Delta = 128$ consistently achieves higher retained performance than $\Delta = 64$. The improvement is especially pronounced at 50% pruning for the two Qwen models: retained performance increases from 64.7% to 69.3% on Qwen3 and from 80.8% to 86.0% on Qwen3.6. At 25% pruning, the corresponding gains are smaller but remain consistent, from 93.1% to 94.8% on Qwen3 and from 96.0% to 97.1% on Qwen3.6. DeepSeek-V2-Lite-Chat shows the same aggregate trend, improving from 67.5% to 69.5% at 25% pruning and from 37.6% to 38.2% at 50% pruning.

These results indicate that a larger tier spacing gives the hierarchical allocator greater freedom to redistribute capacity between experts under the same global channel budget. We therefore use $\Delta = 128$ for the main accuracy experiments. The $\Delta = 64$ checkpoints used in the efficiency study are retained separately because their narrower width range is convenient for the measured width-aware execution setting described in Appendix E.

E HETEROGENEOUS INFERENCE BACKEND AND EFFICIENCY

E.1 COMPACT WIDTH-AWARE EXECUTION

The efficiency study uses the measured $\Delta = 64$, 50%-pruned Qwen3 and Qwen3.6 checkpoints. Qwen3 assigns widths $\{320, 384, 448\}$ to its routed experts, with global counts $\{3056, 32, 3056\}$; Qwen3.6 assigns widths $\{192, 256, 320\}$ with counts $\{5112, 16, 5112\}$. At load time, experts of width K_e are packed into tensors of shape $[E_t, 2K_e, H]$ for the gate–up projection and $[E_t, H, K_e]$ for the down projection. A mapping translates the router's expert identifier into the corresponding width bucket and bucket-local identifier.

The backend chooses its execution path from the number of tokens entering the MoE layer. Token batches smaller than 16 use a slot-wise kernel that reads each selected expert at its true width. Larger batches use homogeneous grouped GEMMs within each width bucket. This hybrid schedule avoids padding while retaining efficient matrix multiplication for prefill and larger decode batches. The router, TopK operation, attention layers, and dense MLPs use the standard vLLM path. CSP assigns every routed expert width 384 on Qwen3 and 256 on Qwen3.6. These homogeneous shapes run directly through the stock vLLM MoE implementation without the width-aware backend.

Table 12: Efficiency measurement protocol.

Item	Setting
Device	1× NVIDIA A100-SXM4-80GB
Serving stack	vLLM 0.23.1, BF16, TP=1
Prompt / output length	2,048 / 128 tokens
Batch size	$\{1, 2, 4, 8, 16\}$
Execution	FULL_AND_PIECEWISE CUDA graphs
Timing	1 warm-up, 3 measured iterations
References	CSP and dense model, stock vLLM

TTFT is measured with one generated token. For a 128-token response, $\mathrm{TPOT} = (t_{\mathrm{E2E}} - t_{\mathrm{TTFT}})/127$. Requests use greedy generation, disabled prefix caching, and disabled detokenization during timing. HARP, CSP, and the dense reference use the same benchmark harness. Compact routed-expert tensors occupy approximately 30.8 versus 56.9 GiB for Qwen3 and 36.5 versus 65.5 GiB for Qwen3.6; vLLM allocates the remaining configured device budget to its KV cache.

Table 13: Accuracy–efficiency at HARP's fastest measured batch for the $\Delta = 64$ system checkpoints. Ret. is mean retained performance (%); E2E is speedup over dense.

		Ret. ↑		E2E ↑	
Model	Batch	CSP	HARP	CSP	HARP
Qwen3	4	59.0	**64.7**	**1.54**×	**1.54**×
Qwen3.6	8	73.9	**80.8**	**1.33**×	**1.33**×

E.2 Raw latency measurements

Tables 14 and 15 provide the measurements underlying Figure 3 and Table 13. Each entry is reported as HARP / CSP / dense. Every measured HARP TTFT and end-to-end latency is lower than the dense reference.

Table 14: Qwen3 latency: HARP / CSP / dense reference.

Batch	TTFT (ms)	TPOT (ms)	E2E (s)
1	81.1 / 79.4 / 111.2	6.53 / 6.10 / 7.17	0.911 / 0.854 / 1.022
2	137.5 / 134.7 / 186.2	7.37 / 7.01 / 9.39	1.074 / 1.025 / 1.378
4	256.1 / 247.6 / 340.3	8.43 / 8.47 / 13.39	1.327 / 1.323 / 2.041
8	511.0 / 494.7 / 679.5	11.28 / 9.49 / 11.32	1.943 / 1.700 / 2.117
16	1028 / 992.3 / 1368	12.34 / 11.26 / 14.82	2.596 / 2.423 / 3.250

Table 15: Qwen3.6 latency: HARP / CSP / dense reference.

Batch	TTFT (ms)	TPOT (ms)	E2E (s)
1	98.0 / 97.7 / 121.1	6.72 / 6.25 / 6.86	0.952 / 0.892 / 0.992
2	158.4 / 160.0 / 196.7	7.54 / 7.05 / 8.14	1.117 / 1.055 / 1.231
4	286.8 / 286.2 / 343.9	9.11 / 8.60 / 10.36	1.443 / 1.379 / 1.660
8	572.7 / 570.2 / 689.4	11.22 / 11.21 / 15.49	1.998 / 1.994 / 2.657
16	1150 / 1148 / 1385	14.73 / 12.89 / 17.71	3.021 / 2.784 / 3.634

F Additional Structural Analysis

We examine the ranked profiles of layers, experts, and channels, and whether high parent scores depend on a few leading children. Every diagnostic uses the structural preservation score in the method,

$$\rho(\Theta) = \mathrm{SP}(\Theta) = \frac{N\|\Theta\|_2^2}{\|\Theta\|_1^2} = 1 + \mathrm{CV}^2(\Theta).$$

The ranking cache stores $\log \rho$, which preserves the ordering. Change points, effect sizes, outlier tests, and removal experiments below are computed on ρ. These measurements characterize the existing allocation rather than set its budgets or width tiers.

F.1 Rank-head diagnostics

We sort each Layer-SP, Expert-SP, and Channel-SP sequence in descending order and search for a mean-shift change point. A candidate boundary must follow the first two ranks, have an adjacent gap at least as large as the sequence's 90th-percentile gap, and have a mean-shift statistic above 1. For channels, the search is confined to the first 25% of ranks. We perturb each sequence 32 times with 0.5% relative noise; a boundary is positionally stable if at least 75% of the trials place it within two ranks of its original position.

Table 16: Rank-head detections using raw SP. Expert heads are counted over MoE layers and channel heads over routed experts.

Model	Expert heads / layers	Channel heads / experts
Qwen3	29/48	2889/6144
Qwen3.6	24/40	4579/10240
DeepSeek-V2-Lite	16/26	636/1664

Expert-rank heads occur in 60–62% of layers across the three models (Table 16), but their strength and stability differ. The representative expert boundary occurs at rank 7 for Qwen3, rank 13 for Qwen3.6, and rank 4 for DeepSeek-V2-Lite. The mean fractions of perturbations retaining an expert boundary within two ranks are 95.6%, 79.6%, and 71.2%, respectively; the last falls below the 75% stability criterion. Channel-head detections cover 47.0%, 44.7%, and 38.2% of routed experts, with representative boundaries at ranks 39, 26, and 71. Their pooled within-two-rank fraction is 100%, although the separation from the tail is comparatively weak. Positional repeatability therefore does not imply a sharp or universal cutoff.

At the layer level, Qwen3 and Qwen3.6 have weak leading separations near ranks 3 and 2, whereas DeepSeek-V2-Lite has a clearer head near rank 4. We select candidate high-ranked layers from the top 10% of each layer ranking when a gap or change point is present: L0, L47, L37, L41, and L25 for Qwen3; L0, L39, L33, and L32 for Qwen3.6; and L11, L10, and L8 for DeepSeek-V2-Lite. A high rank alone does not define a protected set.

F.2 Expert removal and Layer-SP recomputation

We test whether a high layer score is carried by its leading experts. Within each candidate layer, we remove the top expert, top 5%, or top 10% of experts by Expert-SP and recompute $\rho(\Theta_\ell^L \setminus \mathcal{R}_\ell)$ from

the remaining routed-expert weights. This uses the remaining weights' ℓ_1 and ℓ_2 moments rather than an average of surviving Expert-SP scores. Among candidate layers, L0 in Qwen3 and Qwen3.6 and L11 in DeepSeek-V2-Lite exceed the other layers by the auxiliary two-standard-deviation outlier criterion.

Table 17: Layer-SP recomputed after removing leading experts. Top 1 denotes one expert; percentages refer to the expert count.

Model	Layer	Original	Remove top 1	Remove top 5%	Remove top 10%
Qwen3	L0	1.7409	1.7356	1.7055	1.6831
Qwen3.6	L0	2.7268	2.7193	2.6590	2.6073
DeepSeek-V2-Lite	L11	1.5933	1.5928	1.5921	1.5915

After removing the top 10% of experts, these layer scores remain at 96.7%, 95.6%, and 99.9% of their original values, respectively. The elevated Layer-SP is spread across the layer rather than concentrated in its single leading expert. Layer-SP thus captures information distinct from the within-layer Expert-SP ranking.

F.3 Channel removal and Expert-SP recomputation

For each candidate layer, we take its top 10% of experts by Expert-SP: 65 Qwen3, 104 Qwen3.6, and 21 DeepSeek-V2-Lite experts in total. Within each selected expert, we rank coupled gate, up, and down channel slices by Channel-SP. We remove the top 1%, 5%, or 10% of channels and recompute $\rho(\Theta^E_{\ell,e} \setminus \mathcal{C}_{\ell,e})$ from the remaining weights. Table 18 gives representative experts; Figure 5 compares their changes relative to each original score.

Table 18: Expert-SP recomputed after removing leading coupled channels. LxEy denotes expert y in layer x.

Model	Expert	Original	Remove top 1%	Remove top 5%	Remove top 10%
Qwen3	L0E35	2.8486	2.8252	2.7760	2.7172
Qwen3	L47E2	1.7072	1.6826	1.6381	1.6216
Qwen3.6	L0E25	9.1011	9.5048	11.7884	18.1751
DeepSeek-V2-Lite	L11E49	1.6265	1.6241	1.6153	1.6108

The parent response varies across experts. Removing the top 10% of channels reduces Qwen3 L47E2 from 1.7072 to 1.6216 and changes DeepSeek-V2-Lite L11E49 from 1.6265 to 1.6108. For Qwen3.6 L0E25, the recomputed score instead rises from 9.1011 to 18.1751. This follows the definition of ρ: it measures concentration among the remaining parameters, so its value can rise when channels are removed. The intervention measures the dependence of a parent score on its leading children, rather than retained parameter mass.

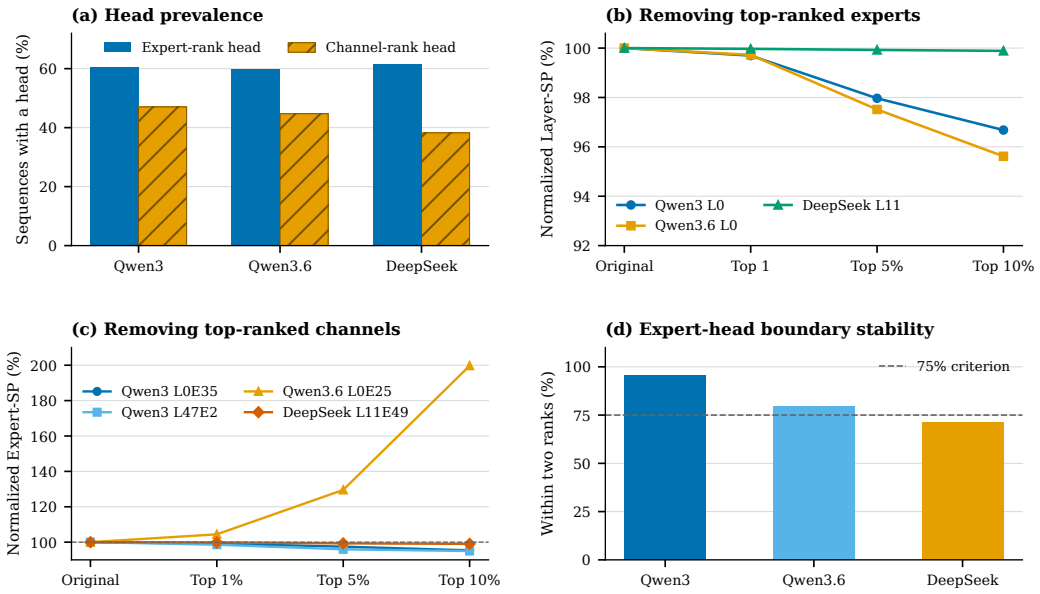

Figure 5: Structural diagnostics using raw SP. (a) Frequency of expert-rank and channel-rank heads. (b) Recomputed Layer-SP after expert removal, normalized by each original layer score. (c) Recomputed Expert-SP after channel removal, normalized by each original expert score. (d) Mean fraction of perturbations placing expert-head boundaries within two ranks of their original positions; the dashed line marks the 75% stability criterion. A normalized score above 100% in (b) or (c) indicates greater concentration after removal.

F.4 Implications for hierarchical allocation

The hierarchy exhibits different forms of concentration at different levels. Layer heads are model dependent, and the leading layer score persists after expert removal. Layer-SP thus supplies a continuous signal for redistributing capacity across layers. Expert-SP reveals recurring within-layer prefixes, with boundary strength and location that vary by model; HARP uses its ordering to assign budget-controlled expert width tiers. Channel-SP orders coupled channels inside each expert, while channel-removal responses show that high Expert-SP can have different channel-level origins. HARP accordingly selects a top-ranked channel prefix at each assigned expert width. The rank-head and removal diagnostics explain these distinct roles without adding a change-point threshold to the allocator.