How to Score Experts for One-Shot MoE Expert Pruning: A Unified Formulation and Selection Principle

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) language models reduce per-token computation through sparse expert activation, yet deployment still requires storing the full expert pool, making one-shot expert pruning a practical approach for reducing memory usage. Although effective, existing criteria are largely heuristic, and no single criterion is universally optimal. Thus, establishing a principle for selecting pruning criteria suited to different deployment objectives remains an important yet largely underexplored problem in one-shot expert pruning. To this end, we introduce a unified formulation for one-shot MoE expert pruning organized around three factors: routing frequency, gate weighting, and activation strength. The formulation yields a criteria selection principle: task-agnostic pruning should favor routed-token-averaged, gate-free activation-based criteria, whereas task-specific pruning can benefit from retaining routing-frequency and gate-weight information. Beyond this principle, the formulation also provides a systematic view of existing heuristic criteria and gives rise to two new task-agnostic criteria, Mean Activation Norm (MAN) and Mean Squared Activation Norm (MSAN). Across four representative MoE models and 16 diverse benchmarks, MAN and MSAN are consistently strong in the task-agnostic setting, obtain the top-two average ranks, and improve average performance by up to 8.8 points over the strongest baseline.

1 Introduction

Mixture-of-Experts (MoE) language models scale capacity through sparse conditional computation, replacing dense feed-forward blocks with expert pools from which a router selects only a few experts per token (Shazeer et al., 2017; Fedus et al., 2022). This mechanism has become a central scaling strategy for recent language models (Jiang et al., 2024; Muennighoff et al., 2024; Liu et al., 2024a; Yang et al., 2025; Baidu, 2025; Meta, 2025; GLM-5-Team et al., 2026; Team et al., 2026; Qwen Team, 2026). However, sparse MoE primarily reduces computation per token, while the memory footprint remains tied to the full expert pool: all experts must still be stored, loaded, and managed during inference. Consequently, memory usage becomes the major deployment bottleneck for MoE models. This motivates a growing line of expert-level compression methods (Li et al., 2023; Lu et al., 2024; Zhang et al., 2025; Lee et al., 2025). Among these, one-shot expert pruning is attractive because it ranks and prunes experts in a single calibration pass, without finetuning, retraining, or extensive combinatorial search (Muzio et al., 2024; Jaiswal et al., 2025; Lasby et al., 2026).

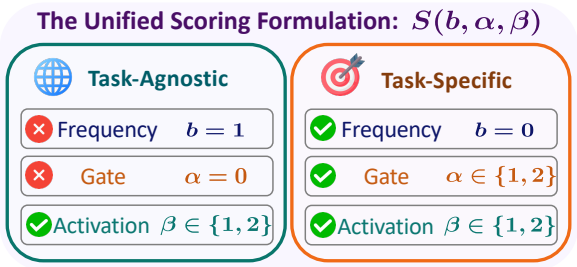

Figure 1: **Selection principle derived from the unified formulation $\mathcal{S}(b, \alpha, \beta)$.** Task-agnostic pruning favors routed-token-averaged, gate-free activation criteria, whereas task-specific pruning can benefit from retaining routing-frequency and gate-weight information.

While one-shot expert pruning is simple and effective, existing methods largely rely on heuristic expert-importance criteria proposed in isolation, and as reflected in Figure 2, no single criterion is consistently optimal across calibration sets, evaluation tasks, and models. This inconsistency likely arises from the fact that one-shot expert pruning is fundamentally a tradeoff. Without retraining or finetuning, preserving experts that support one capability may require discarding experts that support another, especially at high pruning ratio (Liu et al.,

2026a). Moreover, different scoring criteria encode different notions of expert importance. Some are more closely tied to the calibration set, whereas others may capture more stable patterns of expert utility that better transfer across tasks (Zhang et al., 2026). Thus, rather than assuming the existence of a single universally optimal pruning criterion, we identify criterion selection as a central yet under-explored problem in one-shot expert pruning. This raises a natural question: *how should appropriate pruning criteria be selected for different deployment objectives?*

To this end, we first use a single-expert pruning damage measure to identify three core components underlying one-shot expert-pruning criteria: routing frequency, gate weighting, and activation strength. Based on these components, we provide a unified formulation that characterizes how each component affects calibration dependence and derive a criteria selection principle, summarized in Figure 1. This formulation also places existing criteria such as Frequency, SEER (Muzio et al., 2024), EAN (Jaiswal et al., 2025), and REAP (Lasby et al., 2026) into a common design space, explaining why different criteria are suitable for different pruning objectives. Guided by the proposed principle, we further derive two task-agnostic criteria, Mean Activation Norm (MAN) and Mean Squared Activation Norm (MSAN). Across four representative MoE models and 16 downstream benchmarks spanning coding, creative writing, mathematical reasoning, and multiple-choice question answering, the proposed criteria achieve the top-two average ranks and improve average performance by up to 8.8 percentage points over the strongest prior baseline.

Our contributions are summarized as follows:

- By analyzing single-expert pruning damage, we identify three core components for one-shot expert pruning criteria: routing frequency, gate weighting, and activation strength.
- Based on these components we provide a unified scoring formulation for one-shot expert pruning and derive a clear principle for task-agnostic and task-specific pruning.
- Guided by this principle, we derive two task-agnostic criteria, MAN and MSAN. Experiments on four MoE models and 16 diverse benchmarks show that the proposed criteria deliver more balanced overall performance.

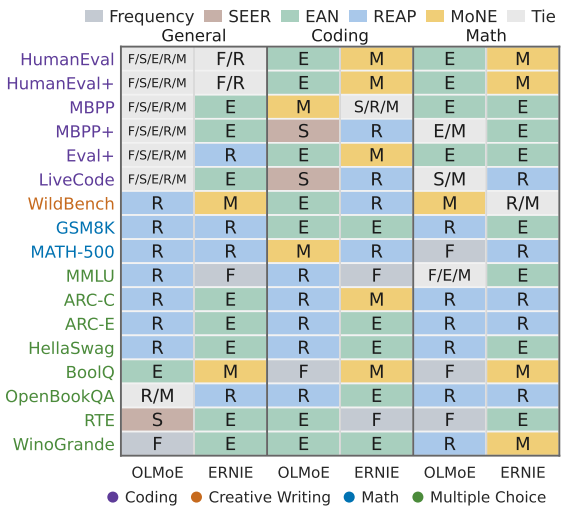

Figure 2: **Winner map for pruning scoring criteria under different calibration sets and models.** Each cell reports the scoring criterion with the highest benchmark score among Frequency, SEER, EAN, REAP, and MoNE for OLMoE-7B at a 25% pruning ratio and ERNIE-4.5-21B at a 50% pruning ratio. Columns group results by calibration set: C4 (General), Evol-CodeAlpaca-v1 (Coding), and Tulu-3-SFT-Personas-Math (Math). No single scoring criterion is universally optimal across these settings.

2 Related Work

2.1 Expert Pruning

Early work on MoE expert pruning considers downstream task specialization and show that substantial expert redundancy can be removed after task-specific fine-tuning (Chen et al., 2022). Koishekenov et al. (2023) study multilingual machine translation and show that pruning language-specific experts improves memory efficiency while largely preserving performance. NAEE (Lu et al., 2024) selects retained experts by minimizing the Frobenius-norm reconstruction error between original and pruned layer outputs. Bai et al. (2025); Liu et al. (2024b, 2026b) formulate pruning as an optimization problem, using gradient-based or gradient-free procedures to optimize expert importance or layer-wise pruning-ratio allocation. EASY-EP (Dong et al., 2025) identifies domain-relevant experts from a few in-domain demonstrations. STUN (Lee et al., 2025) clusters experts by router behaviors to prune or merge redundant ones and then applies unstructured pruning to remaining experts. HodgeCover (Zhong et al., 2026) performs compression using higher-order topological coverage. Beyond these methods, one-shot expert pruning has emerged as a practical and actively stud-

ied approach, owing to its simplicity and effectiveness. Existing methods mainly differ in the expert-importance scoring criterion they adopt. SEER-MoE (Muzio et al., 2024) ranks experts by routing frequency (Frequency) or gate-weighted routing frequency (SEER). Jaiswal et al. (2025) compares different scoring criteria and identify the sum of expert activation norm (EAN) as the strongest. REAP (Lasby et al., 2026) uses gate-weighted activation norms and achieves strong performance on generative tasks, while MoNE (Zhang et al., 2026) uses routing-frequency-weighted activation variance to capture expert redundancy. Despite their practical appeal, existing one-shot expert-pruning methods mainly seek stronger individual scoring criteria. Yet no single scoring criterion consistently dominates, since one-shot pruning inherently trades off different capabilities. Our work differs in taking a unified view of these criteria, relating them through a common score family and characterizing which choices are more suitable for task-specific and task-agnostic pruning.

2.2 Expert Merging

Expert merging compresses MoE models by consolidating multiple experts into fewer experts. MEO (He et al., 2023) performs online token-wise merging by forming a router-score-weighted combination of the activated experts at inference time. Offline methods instead merge experts ahead of deployment: MC-SMoE (Li et al., 2023) first aligns neurons, groups experts using routing information, and merges each group with routing-frequency-weighted averaging; HC-SMoE (Chen et al., 2024) hierarchically clusters experts by output similarity before merging each cluster; and Sub-MoE (Li et al., 2025a) clusters experts and merges them in a shared subspace. REAM (Jha et al., 2026) is a REAP-inspired variant that groups experts and merges their weights instead of pruning them.

2.3 Other Compression Methods

Beyond whole-expert pruning and merging, SMoE models can also be compressed at a finer granularity through quantization (Huang et al., 2024), decomposition-based compression (Gu et al., 2025; He et al., 2025; Li et al., 2025b), and intra-expert weight pruning as in MoE-Pruner (Xie et al., 2024). Some methods also combine expert-level compression with finer-grained reconstruction. DERN (Zhou et al., 2025) first prunes redundant experts and then reallocates neuron-level segments to retained experts. Another line of work combines multiple techniques (He et al., 2024; Liu et al., 2024c). For example, MoE-I^2 (Yang et al., 2024) integrates expert pruning and low-rank decomposition, followed by LoRA fine-tuning (Hu et al., 2022) to recover performance.

3 Preliminaries

Mixture-of-Experts Layer. A Mixture-of-Experts (MoE) layer consists of n feed-forward networks, referred to as experts $\{E_i\}_{i=1}^n$, and a router that activates only the top-k experts for each token (Shazeer et al., 2017; Fedus et al., 2022). For a token t with hidden representation $\mathbf{h}_t \in \mathbb{R}^d$, the router computes logits $\mathbf{z}_t = \mathbf{W}_r \mathbf{h}_t \in \mathbb{R}^n$, where $\mathbf{W}_r \in \mathbb{R}^{n\times d}$ is the router projection matrix. Let $\mathcal{E}_t := \mathrm{TopK}(\mathbf{z}_t, k)$ denote the set of selected experts, where $k \ll n$. The router then normalizes the logits over $\mathcal{E}_t$, yielding sparse gate weights

$$g_{i,t} = \begin{cases} \dfrac{\exp(z_{i,t})}{\sum_{j\in\mathcal{E}_t}\exp(z_{j,t})}, & i \in \mathcal{E}_t, \\ 0, & i \notin \mathcal{E}_t. \end{cases} \tag{1}$$

Let $\mathbf{f}_{i,t} := E_i(\mathbf{h}_t) \in \mathbb{R}^d$ denote the output of expert i for token t. The MoE layer output is

$$\mathbf{y}_t = \textstyle\sum_{i\in\mathcal{E}_t} g_{i,t}\,\mathbf{f}_{i,t}. \tag{2}$$

One-shot expert pruning. Given a calibration set, one-shot expert pruning assigns an importance score to the experts in each layer, ranks them accordingly, and prunes the least important ones to meet a target pruning ratio without any finetuning, retraining, or extensive combinatorial search.

4 Methodology

4.1 Damage Measure for Expert Removal

For an MoE layer, consider the pruned configuration in which the j-th expert E_j and its associated router parameters are removed. The router then computes logits over the remaining experts, followed by top-k selection and gate normalization. Let $\mathcal{E}_t^{(-j)}$ and $\tilde{g}_{i,t}^{(-j)}$ denote the selected expert set and gate weights after this removal, respectively. For a token originally routed to E_j, removing E_j promotes a replacement expert E_r. The original layer output can be decomposed by separating the contribution of E_j from those of the other active experts:

$$\mathbf{y}_t = g_{j,t}\mathbf{f}_{j,t} + \textstyle\sum_{i\in\mathcal{E}_t\setminus\{j\}} g_{i,t}\mathbf{f}_{i,t}. \tag{3}$$

After E_j is removed, the experts that remain active are reweighted, and the replacement expert E_r is added:

$$\mathbf{y}_t^{(-j)} = \sum_{i \in \mathcal{E}_t \setminus \{j\}} \tilde{g}_{i,t}^{(-j)} \mathbf{f}_{i,t} + \tilde{g}_{r,t}^{(-j)} \mathbf{f}_{r,t}. \quad (4)$$

Subtracting Equation (4) from Equation (3) gives the exact output perturbation induced by pruning E_j:

$$\Delta_{j,t} = \mathbf{y}_t - \mathbf{y}_t^{(-j)} = g_{j,t}\mathbf{f}_{j,t} + \boldsymbol{\rho}_{j,t}, \quad (5)$$

where $g_{j,t}\mathbf{f}_{j,t}$ is the direct contribution removed with E_j, and $\boldsymbol{\rho}_{j,t}$ collects the rerouting effects:

$$\boldsymbol{\rho}_{j,t} := \sum_{i \in \mathcal{E}_t \setminus \{j\}} \left(g_{i,t} - \tilde{g}_{i,t}^{(-j)} \right) \mathbf{f}_{i,t} - \tilde{g}_{r,t}^{(-j)} \mathbf{f}_{r,t}, \quad (6)$$

where the first term captures gate renormalization on the surviving experts, and the second term captures the contribution of the promoted replacement expert. Thus, Equation (5) separates the output change into a removed-expert contribution and a rerouting residual.

A scalar damage score can be obtained by measuring the norm of the output perturbation. Given a calibration set with M tokens, the exact single-expert damage can then be written as

$$\widehat{D}_j^{\Delta} := \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \, \|\Delta_{j,t}\|_2. \quad (7)$$

Directly evaluating $\widehat{D}_j^{\Delta}$ requires simulating the removal of each candidate expert E_j: the router logits for E_j must be masked, the top-k set $\mathcal{E}_t^{(-j)}$ must be recomputed, and the selected gates must be renormalized to obtain $\tilde{g}_{i,t}^{(-j)}$. Repeating this procedure for every expert and token is more expensive than a standard calibration pass. In one-shot expert pruning, the importance scores are expected to be collected once on a calibration set and then used directly for ranking, without an expert-wise rerouting procedure. A direct proxy is therefore to ignore the rerouting residual $\boldsymbol{\rho}_{j,t}$ in Equation (5) and measure only the removed contribution $g_{j,t}\mathbf{f}_{j,t}$, giving

$$\widehat{D}_j := \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \, g_{j,t} \|\mathbf{f}_{j,t}\|_2. \quad (8)$$

Empirically, Figure 3 indicates that the expert ranking induced by the proxy damage score is closely aligned with that induced by the exact damage score. This makes $\widehat{D}_j$ a practical choice for one-shot expert pruning, as it avoids the expert-wise recomputation needed to evaluate $\widehat{D}_j^{\Delta}$.

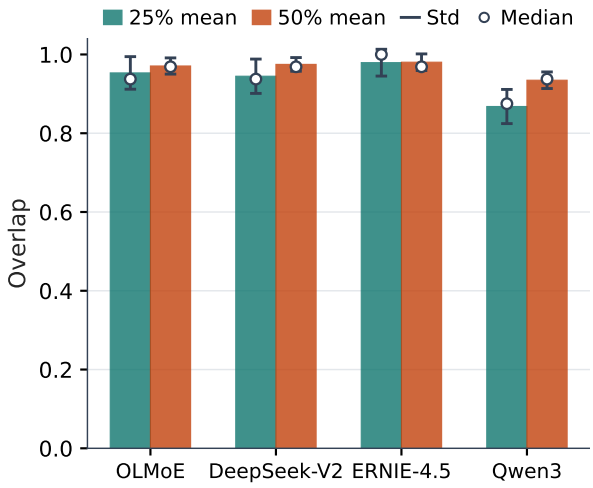

Figure 3: **Expert overlap between the single-expert proxy and exact damage under 25% and 50% pruning ratios.** We use ∼0.5M tokens from C4, and then rank the experts within each layer by the proxy damage score $\widehat{D}_j$ and the exact damage score $\widehat{D}_j^{\Delta}$. Across four distinct and representative models, the overlap is mostly close to or larger than 0.95, and even the lowest case, Qwen3 at 25%, remains approximately above 0.85.

4.2 The Unified Scoring Formulation

Routing frequency, gate weights and activation strength. The proxy damage score in Equation (8) exposes three factors that recur across existing expert-importance scoring criteria. The indicator $\mathbf{1}[j \in \mathcal{E}_t]$ determines which tokens are routed to E_j and therefore carries routing-frequency information. The gate weight $g_{j,t}$ measures how strongly the router assigns weight to E_j on those tokens. The activation term $\|\mathbf{f}_{j,t}\|_2$ measures the strength of the expert output. Because these factors appear multiplicatively in $\widehat{D}_j$, we can ablate or emphasize each factor directly.

The unified formulation. Based on these factors, we define a unified scoring formulation for one-shot expert pruning:

$$S_j(b, \alpha, \beta) := \frac{1}{N_j^b} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \, g_{j,t}^{\alpha} \|\mathbf{f}_{j,t}\|_2^{\beta}, \quad (9)$$

where $N_j := \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t]$ is the number of routed tokens for expert j, and the hyperparameters satisfy $b \in \{0, 1\}$ and $\alpha, \beta \in \{0, 1, 2\}$. Here b determines whether routing frequency is retained (b=0) or converted to a routed-token average (b=1); α is the gate-weight exponent, with α=0 removing gate weighting and larger α making the score more sensitive to gate magnitude by emphasizing routed tokens with larger $g_{j,t}$; and β is the activation exponent, with β=0 giving a purely routing-based criterion, β=1 a norm-like activation score, and β=2 an

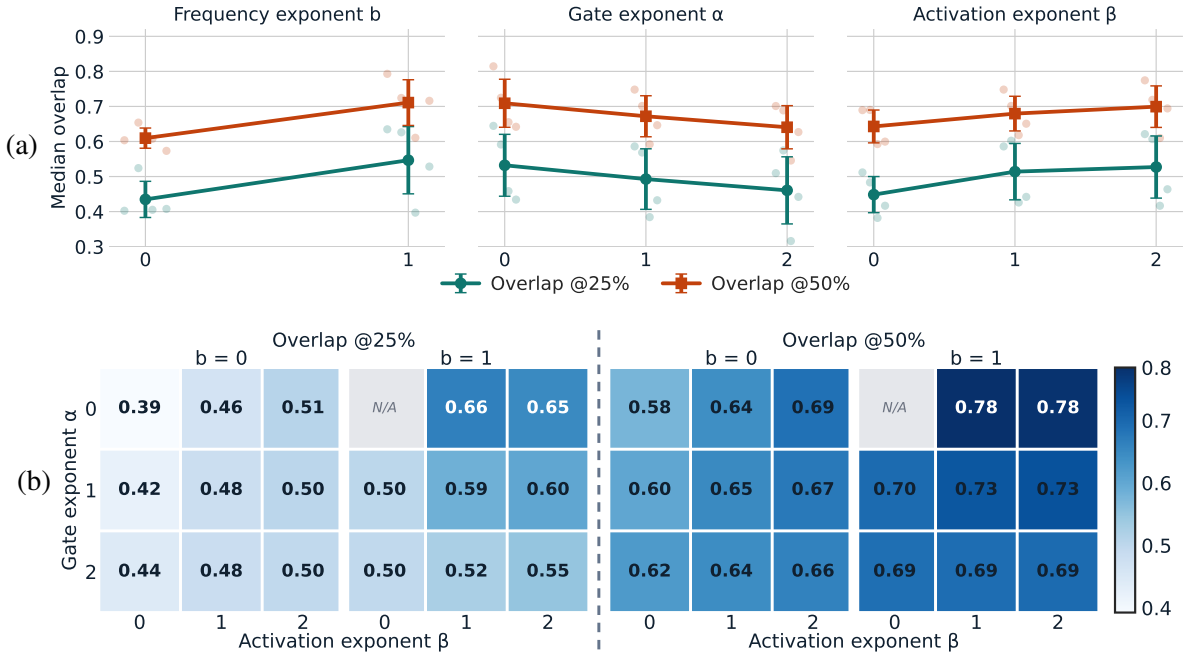

Figure 4: **Overlap of bottom-ranked experts across calibration distributions for different score variants.** Panel (a): Individual component effects. Panel (b): Score-variant trends. For each score variant $S_j(b, \alpha, \beta)$, we compute expert rankings using $\sim$ 0.5M training tokens from each of three calibration sets: C4, Evol-CodeAlpaca-v1, and Tulu-3-SFT-Personas-Math. For each model, we then measure the overlap among the bottom-ranked experts selected for pruning under the three calibration sets at 25% and 50% pruning ratios. N/A denotes the degenerate case $(1, 0, 0)$, which yields a constant score. Results are averaged over the four models: OLMoE-1B-7B-0125-Instruct, DeepSeek-V2-Lite-Chat, ERNIE-4.5-21B-A3B-PT, and Qwen3-30B-A3B-Instruct-2507.

energy-like score. Under this formulation, many existing one-shot expert-pruning scoring criteria can be viewed as special cases, including Frequency $(0, 0, 0)$, SEER $(0, 1, 0)$, EAN $(0, 0, 1)$, and REAP $(1, 1, 1)$. It also yields new criteria obtained by alternative choices.

4.3 Principle for One-shot Expert Pruning

The unified formulation provides a systematic view for choosing expert-pruning scoring criteria under different deployment objectives. We distinguish task-agnostic pruning, which aims to preserve balanced overall performance across heterogeneous capabilities, from task-specific pruning, which aims to preserve or favor a designated capability. To study how each score component relates to these objectives, we use the cross-calibration overlap in Figure 4 as a measure of calibration invariance: if a score identifies nearly the same bottom-ranked experts under different calibration sets, then it is less tied to any single source distribution and is therefore better suited to task-agnostic pruning. As shown in Figure 4 (a), the dominant effect comes from the routing-frequency term. Switching from b=0 to b=1 substantially increases overlap at both pruning ratios, indicating that routing frequency itself is a major source of calibration dependence. For the gate-weight exponent α, smaller values yield higher cross-calibration overlap, with the largest overlap obtained when gate weights are removed, suggesting that gate weights also encode source-specific preferences. By contrast, the role of the activation exponent β is mainly to distinguish routing-only scores from activation-based scores: the main gap is between β=0 and β>0, while the difference between β=1 and β=2 is comparatively small. This leads to a simple principle: task-agnostic pruning should favor gate-free activation scores averaged over routed tokens, while task-specific pruning can benefit from retaining routing-frequency and gate-weight information.

4.4 Mean (Squared) Activation Norm

The heatmaps in Figure 4 (b) instantiate this principle over the full score family. The most calibration-invariant variants concentrate in the $b = 1, \alpha = 0$ block, which corresponds exactly to routed-token mean activation scores that remove routing-frequency scaling and discard gate weights. In particular, $(1, 0, 1)$ and $(1, 0, 2)$ achieve the highest overlap at both 25% and 50% pruning ratios. The model-wise breakdown in Appendix Figure 5

shows the same tendency for the individual models. We denote these variants as Mean Activation Norm (MAN):

$$\mathrm{MAN}_j = \frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \, \|\mathbf{f}_{j,t}\|_2,$$

and Mean Squared Activation Norm (MSAN):

$$\mathrm{MSAN}_j = \frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \, \|\mathbf{f}_{j,t}\|_2^2,$$

respectively. The score definitions used in this paper are summarized in Table 6.

5 Experiments

5.1 Experimental Setup

Models and scoring criteria. We evaluate on four representative and distinct MoE language models: OLMoE-1B-7B-0125-Instruct (Muennighoff et al., 2024), DeepSeek-V2-Lite-Chat (DeepSeek-AI, 2024), ERNIE-4.5-21B-A3B-PT (Baidu, 2025), and Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025). Together, these models cover a broad range of parameter scales (7B–30B), architectural designs (including the presence or absence of shared experts and dense layers), and active experts per token (top-6 and top-8); architectural details are listed in Appendix Table 4. We compare the newly derived scoring criteria from our unified formulation with five representative published baseline criteria: Frequency and SEER (Muzio et al., 2024), Expert Activation Norm (EAN) (Jaiswal et al., 2025), REAP (Lasby et al., 2026), and MoNE (Zhang et al., 2026).

Evaluation suite and calibration data. For task-agnostic pruning, where the target downstream capability is not assumed known, we use C4 (Allen Institute for AI, 2024) as the calibration set, following its common use as a general-domain calibration source in task-agnostic compression (Frantar and Alistarh, 2023; Ling et al., 2024; Xia et al., 2024), and evaluate on 16 downstream benchmarks spanning coding, creative writing, mathematical reasoning, and multiple-choice question answering. For coding, we evaluate on EvalPlus (Liu et al., 2023) and 182 LiveCodeBench (Jain et al., 2024) problems collected between January and April 2025. For creative writing, we evaluate on 146 prompts sampled from WildBench (Lin et al., 2024), using `gpt-oss-120b` (OpenAI, 2025) as the judge. For mathematical reasoning, we evaluate on GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021) with EvalScope (Team, 2024). For multiple-choice question answering, we evaluate on MMLU (Hendrycks et al., 2020), AI2 Reasoning Challenge (ARC-C/ARC-E) (Clark et al., 2018), BoolQ (Clark et al., 2019), OpenBookQA (OBQA) (Mihaylov et al., 2018), HellaSwag (Zellers et al., 2019), Recognizing Textual Entailment (RTE) (Bentivogli et al., 2009), and WinoGrande (WinoG.) (Sakaguchi et al., 2021), all implemented with lm-eval-harness (Gao et al., 2021). For task-specific pruning, we focus on coding and mathematical reasoning, using Evol-CodeAlpaca-v1 (Luo et al., 2023) and Tulu-3-SFT-Personas-Math (Lambert et al., 2024) as the corresponding capability-aligned calibration sets. In all cases, expert-importance scores are computed from 0.5M tokens sampled from the relevant calibration set.

Protocol and hardware. All downstream evaluations are conducted in the zero-shot setting. For open-ended generation benchmarks, we use deterministic decoding with `do_sample=False`; when a temperature parameter is exposed, we set `temperature=0`. This yields greedy, non-sampled generation and improves comparability across pruning methods. All experiments are conducted on NVIDIA L40S 48GB GPUs.

5.2 Results

Task-agnostic results. Table 1 evaluates one-shot pruning under the task-agnostic setting, where expert-importance scores are computed on C4 and the pruned models are evaluated across coding, writing, math, and multiple-choice benchmarks; the full per-benchmark and 50% ratio results and are reported in Table 5. The main trend is clear: the routed-token-averaged, gate-free mean-activation scores, MAN $(1, 0, 1)$ and MSAN $(1, 0, 2)$, are the strongest overall choices. Across all four representative MoE models, either MAN or MSAN achieves the best overall average and the best average rank. Relative to the strongest prior baseline in each model, the best mean-activation criterion improves the overall average by $+1.3$, $+2.8$, $+2.9$, and $+8.8$ percentage points on OLMoE, DeepSeek, ERNIE, and Qwen3, respectively. These gains are obtained without task-specific calibration or recovery finetuning. The per-capability results further clarify why MAN and MSAN are preferable for task-agnostic pruning. Their advantage is not merely an artifact of one benchmark group: on ERNIE, MAN/MSAN improve the overall average

Table 1: **Task-agnostic pruning with C4 calibration.** We report results at a 25% pruning ratio for four representative MoE models, comparing published baseline scoring criteria with the proposed mean-activation criteria MAN and MSAN (blue). Benchmarks are grouped by capability; Avg and Avg Rank summarize balanced performance across the displayed benchmark columns. Bold and underline indicate the best and second-best results within each block.

Model	Pruning Ratio	Criterion	(b, α, β)	Coding Eval+	Coding LiveCode	Writing WildBench	Math GSM8K	Math MATH-500	MC MC Avg	Overall Avg	Overall Avg Rank
OLMoE	0%	Full	-	0.341	0.033	0.444	0.682	0.222	0.653	0.396	-
	25%	Frequency	$(0, 0, 0)$	0.000	0.000	0.127	0.033	0.024	0.560	0.124	5.67
		SEER	$(0, 1, 0)$	0.000	0.000	0.141	0.037	0.012	0.564	0.126	5.42
		EAN	$(0, 0, 1)$	0.000	0.000	0.184	0.133	0.012	0.582	0.152	4.58
		REAP	$(1, 1, 1)$	0.000	0.000	**0.263**	0.139	0.036	**0.601**	0.173	2.83
		MoNE	-	0.000	0.000	0.181	0.117	0.006	0.583	0.148	5.00
		MAN	$(1, 0, 1)$	**0.009**	0.000	0.260	**0.208**	0.046	0.594	**0.186**	**2.00**
		MSAN	$(1, 0, 2)$	0.008	0.000	0.242	0.194	**0.056**	0.589	0.181	2.50
DeepSeek	0%	Full	-	0.549	0.104	0.418	0.610	0.298	0.678	0.443	-
	25%	Frequency	$(0, 0, 0)$	0.000	0.000	0.291	0.023	0.012	0.602	0.155	5.33
		SEER	$(0, 1, 0)$	0.000	0.000	0.155	0.034	0.016	0.602	0.134	5.67
		EAN	$(0, 0, 1)$	0.000	0.000	**0.295**	0.312	0.028	0.621	0.209	3.33
		REAP	$(1, 1, 1)$	0.007	0.000	0.174	0.281	0.028	0.620	0.185	3.92
		MoNE	-	0.000	0.000	0.200	0.227	0.024	0.629	0.180	4.42
		MAN	$(1, 0, 1)$	0.001	0.000	0.154	0.287	0.032	**0.636**	0.185	3.33
		MSAN	$(1, 0, 2)$	**0.025**	0.000	0.238	**0.428**	**0.102**	0.630	**0.237**	**2.00**
ERNIE	0%	Full	-	0.867	0.247	0.479	0.829	0.780	0.721	0.654	-
	25%	Frequency	$(0, 0, 0)$	0.254	0.055	0.352	0.647	0.316	0.655	0.380	6.67
		SEER	$(0, 1, 0)$	0.256	0.060	0.381	0.748	0.368	0.658	0.412	5.25
		EAN	$(0, 0, 1)$	0.300	0.055	0.408	0.673	0.370	0.682	0.415	4.50
		REAP	$(1, 1, 1)$	0.277	0.060	0.414	0.760	0.528	0.700	0.456	3.25
		MoNE	-	0.217	0.055	**0.419**	0.774	0.384	0.678	0.421	4.33
		MAN	$(1, 0, 1)$	0.343	**0.077**	0.402	0.801	**0.580**	**0.705**	**0.485**	**2.00**
		MSAN	$(1, 0, 2)$	**0.354**	0.066	0.404	**0.813**	0.564	0.704	0.484	**2.00**
Qwen3	0%	Full	-	0.871	0.368	0.644	0.923	0.802	0.737	0.724	-
	25%	Frequency	$(0, 0, 0)$	0.000	0.000	**0.632**	0.904	0.196	0.732	0.411	4.83
		SEER	$(0, 1, 0)$	0.003	0.000	0.612	0.913	0.202	0.734	0.411	3.92
		EAN	$(0, 0, 1)$	0.001	0.000	0.623	0.910	0.194	**0.735**	0.411	4.42
		REAP	$(1, 1, 1)$	0.599	0.137	0.600	0.879	0.778	0.721	0.619	4.67
		MoNE	-	0.000	0.000	0.627	0.911	0.206	0.733	0.413	4.17
		MAN	$(1, 0, 1)$	**0.868**	**0.346**	0.570	**0.937**	0.792	0.727	**0.707**	**2.75**
		MSAN	$(1, 0, 2)$	0.847	0.335	0.559	0.935	**0.796**	0.727	0.700	3.25

while also giving the strongest or near-strongest results on coding, math, and multiple-choice evaluation. On Qwen3, the gap is particularly large for coding: under C4 calibration, Frequency, SEER, EAN, and MoNE nearly collapse on EvalPlus and LiveCodeBench, whereas MAN preserves coding performance close to the full model. Using routed-token averages and removing gate weighting makes MAN/MSAN more reliable across diverse downstream evaluations, which explains their consistently stronger average performance and ranking.

Task-specific pruning with capability-aligned calibration. Table 2 evaluates task-specific pruning, where expert-importance scores for coding are computed on Evol-CodeAlpaca-v1 and scores for math are computed on Tulu-3-SFT-Personas-Math. Unlike the task-agnostic setting, the calibration distribution is intentionally aligned with the target capability. In this case, routing-frequency and gate-weight signals become informative rather than nuisances, and the strongest scoring criteria tend to combine all three factors in our formulation: routing frequency, gate magnitude, and activation strength. This trend is visible when comparing MAN/MSAN with the routing-frequency-retaining, gate-weighted activation criteria. MAN and MSAN use routed-token averages and remove gate weighting, which makes them strong for task-agnostic pruning but less consistently optimal here. By contrast, $(0, 1, 1)$ and $(0, 2, 2)$ retain routing frequency, gate weighting, and activation information. They achieve the best average rank on OLMoE at a 25% pruning ratio, ERNIE at a 25% pruning ratio, and Qwen3 at a 50% pruning ratio, while remaining competitive in most other settings. The main exceptions are DeepSeek, where REAP is strongest, and Qwen3 at a 25% pruning ratio, where EAN achieves the best average rank. Overall, the capability-aligned calibration results support the task-specific side of our principle: when pruning

Table 2: **Task-specific pruning with capability-aligned calibration.** For coding, expert-importance scores are computed on Evol-CodeAlpaca-v1; for math, scores are computed on Tulu-3-SFT-Personas-Math. We compare published baseline scoring criteria and MAN/MSAN with routing-frequency-retaining and gate-weighted criteria from our unified scoring formulation, which are in principle suited to task-specific pruning (blue).

Model	Pruning Ratio	Criterion	Coding		Math		Rank	Model	Coding		Math		Rank
			Eval+	LiveCode	GSM8K	MATH-500	Avg Rank		Eval+	LiveCode	GSM8K	MATH-500	Avg Rank
OLMoE	0%	Full	0.341	0.033	0.682	0.222	-	DeepSeek	0.549	0.104	0.610	0.298	-
	25%	Frequency	0.341	0.022	0.593	0.226	3.75		0.297	**0.099**	0.476	0.188	6.62
		SEER	0.339	**0.027**	0.594	0.194	4.88		0.404	**0.099**	0.477	0.196	5.75
		EAN	0.345	0.016	0.622	0.210	3.50		0.440	0.088	0.479	0.172	6.00
		REAP	0.300	0.011	**0.626**	0.198	5.62		**0.465**	0.088	**0.585**	0.218	**2.00**
		MoNE	0.339	0.011	0.618	0.198	5.75		0.406	0.088	0.581	0.210	4.25
		MAN	0.243	0.016	0.541	0.216	6.62		0.444	0.077	0.551	0.196	5.12
		MSAN	0.254	**0.027**	0.558	0.180	6.75		0.411	0.082	0.561	**0.226**	4.38
		(0,1,1)	0.333	**0.027**	0.619	**0.246**	**3.00**		0.435	0.066	0.500	0.186	7.00
		(0,2,2)	**0.348**	0.011	0.581	0.210	5.12		0.442	0.082	0.578	0.216	3.88
ERNIE	0%	Full	0.867	0.247	0.829	0.780	-	Qwen3	0.871	0.368	0.923	0.802	-
	25%	Frequency	0.818	0.181	**0.832**	0.736	7.25		0.862	0.357	0.897	0.794	6.00
		SEER	0.830	0.214	0.822	0.768	6.50		0.851	0.363	0.898	**0.806**	6.00
		EAN	0.821	**0.231**	0.825	0.774	5.75		0.864	**0.396**	**0.917**	0.792	**2.62**
		REAP	0.835	**0.231**	0.829	0.792	3.38		0.852	0.335	0.904	**0.806**	6.25
		MoNE	0.826	0.209	0.830	**0.800**	5.12		0.859	0.363	0.914	0.780	5.62
		MAN	0.830	0.214	**0.832**	0.776	4.25		**0.871**	**0.396**	0.904	0.788	4.00
		MSAN	0.835	0.209	**0.832**	0.748	5.00		0.860	0.363	0.905	0.776	6.25
		(0,1,1)	0.834	**0.231**	0.827	0.760	5.00		0.868	0.379	0.912	0.792	3.38
		(0,2,2)	**0.837**	0.225	0.831	0.796	**2.75**		0.860	0.363	0.912	0.792	4.88
	50%	Frequency	0.647	0.143	0.729	0.554	9.00		0.700	0.225	0.858	0.764	7.88
		SEER	0.698	0.170	0.766	0.634	6.00		0.700	0.247	0.857	0.766	7.62
		EAN	0.657	0.148	0.801	0.622	6.00		0.841	0.313	0.874	**0.800**	4.25
		REAP	0.737	0.198	0.765	**0.680**	3.50		0.835	**0.357**	0.884	0.740	4.50
		MoNE	**0.741**	0.176	0.785	0.616	4.75		0.842	0.346	0.872	0.780	4.62
		MAN	0.716	**0.214**	0.790	0.676	**2.62**		0.850	0.346	0.861	0.750	5.38
		MSAN	0.692	0.176	0.785	0.632	5.75		0.832	0.324	0.856	0.782	6.50
		(0,1,1)	0.706	0.203	0.790	0.604	4.62		**0.855**	0.352	0.883	0.794	**2.12**
		(0,2,2)	0.733	0.192	**0.804**	0.654	2.75		0.854	0.352	**0.892**	0.784	**2.12**

for a known target capability, expert importance should reflect not only how strongly an expert activates, but also how often and how confidently the router uses it on matched data. Thus, task-specific pruning benefits from retaining routing-frequency, gate-weight, and activation signals jointly, whereas MAN/MSAN are better suited to the task-agnostic setting where routing effects should be suppressed.

Loading memory. Table 3 shows that at a 50% pruning ratio, expert pruning reduces loading memory by ∼50% across the evaluated MoE models, validating it as a practical way to address memory usage, the main bottleneck in MoE deployment.

Table 3: **Model loading memory before and after pruning.** Models are loaded in bf16. GPU denotes the number of L40S GPUs needed to load the model and perform one-shot pruning. Loading memory is reported as after / before pruning at 50% pruning ratio.

Model	GPU	Loading Memory (GB)
OLMoE-7B	1×L40S	6.89 / 12.89
DeepSeek-16B	1×L40S	15.86 / 29.27
ERNIE-21B	2×L40S	21.67 / 40.66
Qwen3-30B	2×L40S	29.93 / 56.92

6 Conclusion

We focus on selecting an expert-importance criterion suited to a given deployment objective for one-shot MoE expert pruning, rather than identifying a universally dominant score. From a single-expert removal analysis, we derived a unified formulation that decomposes scoring criteria into routing frequency, gate weighting, and activation strength. This formulation exposes a criteria selection principle: task-agnostic pruning should suppress calibration-specific routing effects by using routed-token-averaged, gate-free activation criteria, whereas task-specific pruning can benefit from routing-frequency and gate-weight signals when calibration data are aligned with the target capability. This perspective explains why routing-only criteria such as Frequency and SEER are sensitive to calibration distributions, and why activation-based criteria can be adapted to different objectives through explicit choices about routed-token averaging and gate weighting. It also yields MAN and MSAN, two new task-agnostic criteria that better preserve balanced performance across heterogeneous benchmarks, improving the average by up to 8.8 percentage points over the strongest prior baseline. Overall, our results provide a compact and empirically supported basis for choosing one-shot expert-pruning criteria under different deployment requirements while keeping the resulting method simple and broadly applicable in practice.

Limitations

Our unified formulation covers commonly used one-shot expert-pruning criteria including Frequency, SEER, EAN, and REAP. Although the principle suggested by this formulation—that routing-related signals introduce calibration dependence and should therefore be suppressed for task-agnostic pruning while retained when the calibration data are target-aligned—may extend beyond this exact score family, not all criteria can be represented in the same form. For example, MoNE uses routing-frequency-weighted activation variance across routed tokens to characterize expert redundancy. Although its routing-frequency component is related to our formulation and the resulting principle may partially transfer, the variance statistic itself is not fully captured by $\mathcal{S}(b, \alpha, \beta)$. In addition, our study focuses on expert pruning, where experts are removed according to importance scores; expert merging modifies multiple experts jointly and may require a different formulation and selection principle.

References

Allen Institute for AI. 2024. allenai/c4 · datasets at Hugging Face. https://huggingface.co/datasets/allenai/c4.

Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. 2025. Diep: Adaptive mixture-of-experts compression through differentiable expert pruning. *Preprint*, arXiv:2509.16105.

Baidu. 2025. Ernie 4.5 technical report. Technical report, Baidu. Technical report.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

I Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, Chun-Yi Lee, and 1 others. 2024. Retraining-free merging of sparse moe via hierarchical clustering. *arXiv preprint arXiv:2410.08589*.

Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

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

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

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, and 1 others. 2021. A framework for few-shot language model evaluation. *Zenodo*.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Hao Gu, Wei Li, Lujun Li, Qiyuan Zhu, Mark Lee, Shengjie Sun, Wei Xue, and Yike Guo. 2025. Delta decompression for moe-based llms compression. *arXiv preprint arXiv:2502.17298*.

Shwai He, Daize Dong, Liang Ding, and Ang Li. 2024. Towards efficient mixture of experts: A holistic study of compression techniques. *arXiv preprint arXiv:2406.02500*.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. 2023. Merging experts into one: Improving computational efficiency of mixture of experts. *arXiv preprint arXiv:2310.09832*.

Yifei He, Yang Liu, Chen Liang, and Hany Hassan Awadalla. 2025. Efficiently editing mixture-of-experts models with compressed experts. *arXiv preprint arXiv:2503.00634*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. 2024. Mixture compressor for mixture-of-experts llms gains more. *arXiv preprint arXiv:2410.06270*.

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

Ajay Jaiswal, Jianyu Wang, Yixiao Li, Pingzhi Li, Tianlong Chen, Zhangyang Wang, Chong Wang, Ruoming Pang, and Xianzhi Du. 2025. Finding fantastic experts in moes: A unified study for expert dropping strategies and observations. *arXiv preprint arXiv:2504.05586*.

Saurav Jha, Maryam Hashemzadeh, Ali Saheb Pasand, Ali Parviz, Min-Joong Lee, and Boris Knyazev. 2026. Ream: Merging improves pruning of experts in llms. *Preprint*, arXiv:2604.04356.

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, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yeskendir Koishekenov, Alexandre Berard, and Vassilina Nikoulina. 2023. Memory-efficient nllb-200: Language-specific expert pruning of a massively multilingual machine translation model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3567–3585.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. 2026. REAP the experts: Why pruning prevails for one-shot moe compression. In *The Fourteenth International Conference on Learning Representations*.

Jaeseong Lee, Seung-won Hwang, Aurick Qiao, Daniel F Campos, Zhewei Yao, and Yuxiong He. 2025. Stun: Structured-then-unstructured pruning for scalable moe pruning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13660–13676.

Lujun Li, Zhu Qiyuan, Jiacheng Wang, Wei Li, Hao Gu, Sirui Han, and Yike Guo. 2025a. Sub-moe: Efficient mixture-of-expert llms compression via subspace expert merging. *arXiv preprint arXiv:2506.23266*.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *arXiv preprint arXiv:2310.01334*.

Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G Lee, Shengjie Sun, Wei Xue, and Yike Guo. 2025b. Moe-svd: Structured mixture-of-experts llms compression via singular value decomposition. In *International Conference on Machine Learning*, pages 35209–35230. PMLR.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.

Gui Ling, Ziyang Wang, Yuliang Yan, and Qingwen Liu. 2024. Slimgpt: Layer-wise structured pruning for large language models. *Advances in Neural Information Processing Systems*, 37:107112–107137.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Enshu Liu, Junyi Zhu, Zinan Lin, Xuefei Ning, Matthew B Blaschko, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024b. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. *arXiv preprint arXiv:2407.00945*.

Jiacheng Liu, Peng Tang, Wenfeng Wang, Yuhang Ren, Xiaofeng Hou, Pheng-Ann Heng, Minyi Guo, and Chao Li. 2024c. A survey on inference optimization techniques for mixture of experts models. *arXiv preprint arXiv:2412.14219*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language

models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572.

Zongfang Liu, Shengkun Tang, Yifan Shen, Huan Wang, and Xin Yuan. 2026a. Aimer: Calibration-free task-agnostic moe pruning. *Preprint*, arXiv:2603.18492.

Zongfang Liu, Shengkun Tang, Boyang Sun, Zhiqiang Shen, and Xin Yuan. 2026b. Evoesap: Non-uniform expert pruning for sparse moe. *arXiv preprint arXiv:2603.06003*.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. *arXiv preprint arXiv:2402.14800*.

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

AI Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on*, 4(7):2025.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, and 1 others. 2024. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*.

Alexandre Muzio, Alex Sun, and Churan He. 2024. Seer-moe: Sparse expert efficiency through regularization for mixture-of-experts. *arXiv preprint arXiv:2404.05089*.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, and 307 others. 2026. Kimi k2.5: Visual agentic intelligence. *Preprint*, arXiv:2602.02276.

ModelScope Team. 2024. EvalScope: Evaluation framework for large models.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared llama: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations*, volume 2024, pages 5385–5409.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. 2024. Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. 2024. Moe-i^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. *arXiv preprint arXiv:2411.01016*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Geng Zhang, Yuxuan Han, Yuxuan Lou, Yiqi Zhang, Wangbo Zhao, and Yang You. 2026. Mone: Replacing redundant experts with lightweight novices for structured pruning of moe. *Preprint*, arXiv:2507.00390.

Zeliang Zhang, Xiaodong Liu, Hao Cheng, Chenliang Xu, and Jianfeng Gao. 2025. Diversifying the expert knowledge for task-agnostic pruning in sparse mixture-of-experts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 86–102.

Tao Zhong, Dongzhe Zheng, and Christine Allen-Blanchette. 2026. Hodgecover: Higher-order topological coverage drives compression of sparse mixture-of-experts. *arXiv preprint arXiv:2605.13997*.

Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, Hehe Fan, and 1 others. 2025. Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts llms. *arXiv preprint arXiv:2509.10377*.

A Model Architecture Details

Table 4: **Architectural diversity of the evaluated MoE models.**

Model	Total params	Experts per layer	Active experts per token	Shared experts	Dense/MoE layout
OLMoE-1B-7B-0125-Instruct	7B	64	top-8	No	0 dense, 16 MoE layers
DeepSeek-V2-Lite-Chat	16B	64 routed	top-6	Yes (2 shared)	1 dense, 26 MoE layers
ERNIE-4.5-21B-A3B-PT	21B	64 routed	top-6	Yes (2 shared)	1 dense, 27 MoE layers
Qwen3-30B-A3B-Instruct-2507	30B	128	top-8	No	0 dense, 48 MoE layers

B Model-wise Calibration Overlap

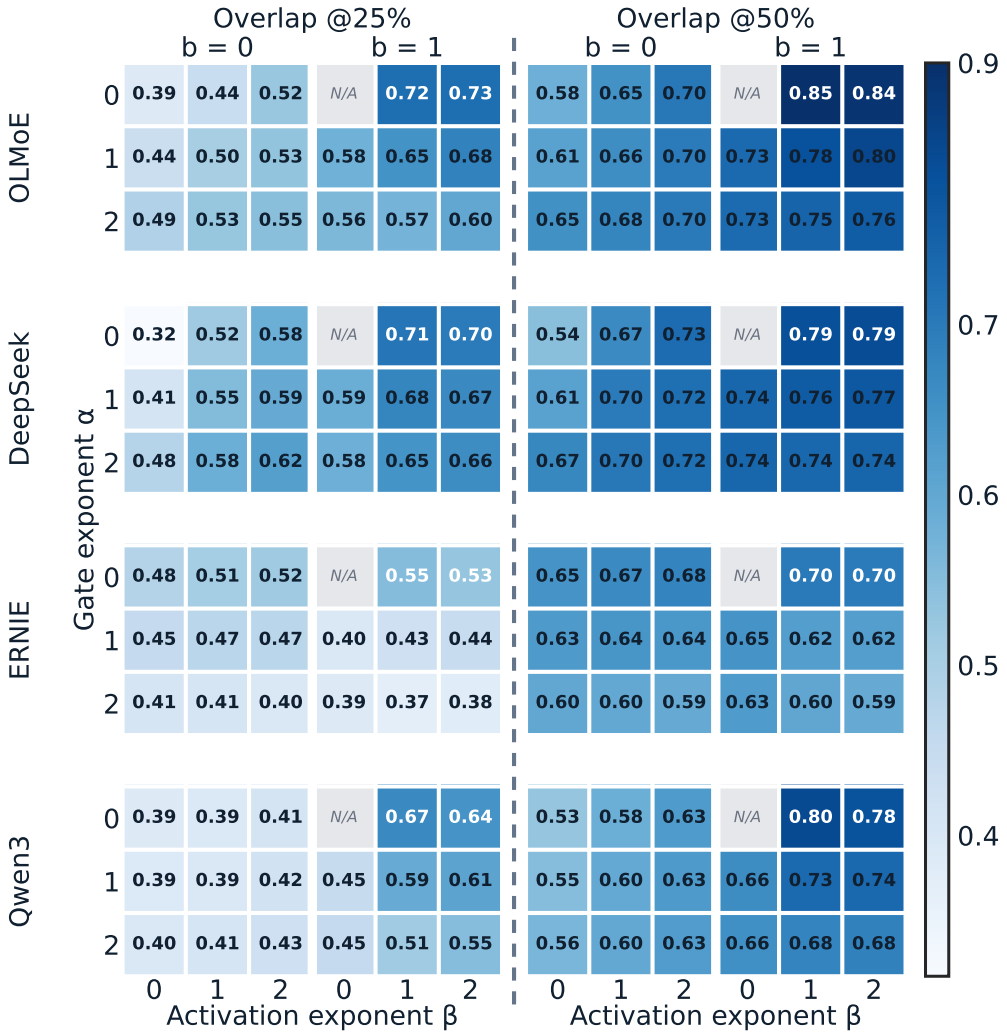

Figure 5: **Model-wise overlap of bottom-ranked experts across calibration distributions.** Each cell reports the overlap among bottom-ranked experts selected by a score variant $S_j(b, \alpha, \beta)$ when rankings are computed from C4, Evol-CodeAlpaca-v1, and Tulu-3-SFT-Personas-Math. Results are shown separately for each model at 25% and 50% pruning ratios. N/A denotes the degenerate case $(1, 0, 0)$. The averaged summary is shown in Figure 4.

C Detailed Task-Agnostic Results

Table 5: Task-agnostic pruning comparison on C4 with individual benchmark results. OLMoE and DeepSeek are reported at a 25% pruning ratio; ERNIE and Qwen3 are reported at 25% and 50% pruning ratios. HumanEval and HumanEval+ are abbreviated as HEVAL and HEVAL+, OpenBookQA as OBQA, and WinoGrande as WinoG. HEVAL, HEVAL+, MBPP, and MBPP+ are the Eval+ sub-benchmarks.

				Coding					Writing	Math		MC							
Model	**Pruning Ratio**	**Criterion**	(b, α, β)	**HEVAL**	**HEVAL+**	**MBPP**	**MBPP+**	**LiveCode**	**WildBench**	**GSM8K**	**MATH-500**	**ARC-C**	**ARC-E**	**BoolQ**	**HellaSwag**	**MMLU**	**OBQA**	**RTE**	**WinoG**
OLMoE	0%	Full	-	0.354	0.323	0.373	0.312	0.033	0.444	0.682	0.222	0.490	0.758	0.766	0.808	0.534	0.470	0.711	0.684
	25%	Frequency	(0, 0, 0)	0.000	0.000	0.000	0.000	0.000	0.127	0.033	0.024	0.382	0.582	0.691	0.727	0.325	0.416	0.690	**0.665**
		SEER	(0, 1, 0)	0.000	0.000	0.000	0.000	0.000	0.141	0.037	0.012	0.388	0.590	0.694	0.733	0.320	0.414	**0.711**	0.661
		EAN	(0, 0, 1)	0.000	0.000	0.000	0.000	0.000	0.184	0.133	0.012	0.454	0.656	**0.736**	0.752	0.344	0.422	0.632	0.661
		REAP	(1, 1, 1)	0.000	0.000	0.000	0.000	0.000	**0.263**	0.139	0.036	**0.487**	**0.720**	0.695	**0.760**	**0.397**	**0.436**	0.661	0.654
		MoNE	-	0.000	0.000	0.000	0.000	0.000	0.181	0.117	0.006	0.446	0.661	0.727	0.751	0.351	**0.436**	0.639	0.653
		MAN	(1, 0, 1)	**0.012**	**0.012**	0.005	0.005	0.000	0.260	**0.208**	0.046	0.469	0.683	0.719	**0.760**	0.395	**0.436**	0.639	0.653
		MSAN	(1, 0, 2)	0.006	0.006	**0.011**	**0.011**	0.000	0.242	0.194	**0.056**	0.457	0.669	0.692	0.755	0.395	**0.436**	0.664	0.646
DeepSeek	0%	Full	-	0.591	0.524	0.585	0.497	0.104	0.418	0.610	0.298	0.541	0.785	0.829	0.808	0.567	0.456	0.726	0.712
	25%	Frequency	(0, 0, 0)	0.000	0.000	0.000	0.000	0.000	0.291	0.023	0.012	0.428	0.667	0.729	0.751	0.417	0.424	**0.726**	0.671
		SEER	(0, 1, 0)	0.000	0.000	0.000	0.000	0.000	0.155	0.034	0.016	0.451	0.680	0.722	0.754	0.394	0.422	**0.726**	0.669
		EAN	(0, 0, 1)	0.000	0.000	0.000	0.000	0.000	**0.295**	0.312	0.028	0.474	0.694	0.747	0.779	0.456	0.430	0.690	0.703
		REAP	(1, 1, 1)	0.006	0.006	0.008	0.008	0.000	0.174	0.281	0.028	0.483	0.730	0.688	**0.780**	**0.466**	0.430	0.690	0.696
		MoNE	-	0.000	0.000	0.000	0.000	0.000	0.200	0.227	0.024	0.479	0.688	0.750	0.779	0.454	**0.462**	**0.726**	0.691
		MAN	(1, 0, 1)	0.000	0.000	0.003	0.003	0.000	0.154	0.287	0.032	0.492	**0.752**	0.764	0.778	0.454	0.446	0.693	**0.704**
		MSAN	(1, 0, 2)	**0.012**	**0.012**	**0.040**	**0.037**	0.000	0.238	**0.428**	**0.102**	**0.503**	0.734	**0.765**	0.778	0.452	0.446	0.661	0.703
ERNIE	0%	Full	-	0.909	0.878	0.915	0.765	0.247	0.479	0.829	0.780	0.564	0.782	0.872	0.814	0.739	0.462	0.816	0.717
	25%	Frequency	(0, 0, 0)	0.201	0.165	0.360	0.288	0.055	0.352	0.647	0.316	0.518	0.727	0.849	0.719	0.571	0.390	0.791	0.679
		SEER	(0, 1, 0)	0.232	0.195	0.317	0.278	0.060	0.381	0.748	0.368	0.511	0.750	0.845	0.736	0.599	0.400	0.747	0.676
		EAN	(0, 0, 1)	0.299	0.262	0.347	0.294	0.055	0.408	0.673	0.370	0.535	0.750	0.840	**0.790**	0.597	0.436	0.791	**0.716**
		REAP	(1, 1, 1)	0.244	0.232	0.341	0.291	0.060	0.414	0.760	0.528	**0.577**	0.791	0.851	0.775	0.645	0.444	0.816	0.704
		MoNE	-	0.201	0.177	0.257	0.233	0.055	**0.419**	0.774	0.384	0.542	0.748	0.864	0.777	0.540	0.442	0.812	0.701
		MAN	(1, 0, 1)	0.293	0.262	**0.437**	**0.381**	**0.077**	0.402	0.801	**0.580**	0.573	**0.798**	0.868	0.776	0.661	**0.478**	0.794	0.688
		MSAN	(1, 0, 2)	**0.341**	**0.305**	0.410	0.360	0.066	0.404	**0.813**	0.564	0.562	0.793	**0.874**	0.773	**0.674**	0.458	**0.820**	0.678
	50%	Frequency	(0, 0, 0)	0.012	0.012	0.003	0.003	0.000	0.164	0.051	0.018	0.386	0.583	0.740	0.579	0.488	0.324	0.733	0.625
		SEER	(0, 1, 0)	0.000	0.000	0.005	0.005	0.000	0.174	0.106	0.020	0.386	0.599	0.711	0.573	0.460	0.324	0.675	0.613
		EAN	(0, 0, 1)	0.006	0.006	0.013	0.013	0.005	0.256	0.080	0.026	0.446	0.675	0.732	**0.680**	0.470	0.394	**0.762**	**0.698**
		REAP	(1, 1, 1)	0.012	0.012	0.011	0.008	0.000	0.237	0.478	0.158	0.402	0.596	0.736	0.666	0.394	**0.398**	0.715	0.676
		MoNE	-	0.006	0.006	0.003	0.003	0.000	0.257	0.174	0.048	0.381	0.599	**0.802**	0.677	0.376	0.390	0.737	0.682
		MAN	(1, 0, 1)	**0.043**	**0.043**	0.077	0.066	**0.022**	0.273	0.672	0.190	0.490	0.714	0.797	0.677	**0.543**	0.386	0.733	0.664
		MSAN	(1, 0, 2)	0.037	0.037	**0.093**	**0.085**	0.011	**0.285**	**0.681**	**0.200**	**0.526**	**0.737**	0.691	**0.680**	0.500	0.382	0.675	0.668
Qwen3	0%	Full	-	0.939	0.902	0.892	0.751	0.368	0.644	0.923	0.802	0.625	0.838	0.887	0.797	0.802	0.446	0.769	0.736
	25%	Frequency	(0, 0, 0)	0.000	0.000	0.000	0.000	0.000	**0.632**	0.904	0.196	0.625	0.839	0.886	**0.795**	0.761	0.442	0.773	0.732
		SEER	(0, 1, 0)	0.006	0.006	0.000	0.000	0.000	0.612	0.913	0.202	0.630	0.845	**0.888**	**0.795**	0.760	0.446	0.776	0.732
		EAN	(0, 0, 1)	0.000	0.000	0.003	0.003	0.000	0.623	0.910	0.194	**0.642**	**0.854**	0.885	**0.795**	**0.768**	0.442	0.765	0.731
		REAP	(1, 1, 1)	0.585	0.549	0.683	0.579	0.137	0.600	0.879	0.778	0.620	0.825	0.885	0.776	0.754	0.426	0.751	0.729
		MoNE	-	0.000	0.000	0.000	0.000	0.000	0.627	0.911	0.206	0.632	0.851	0.886	**0.795**	0.766	0.438	0.765	**0.734**
		MAN	(1, 0, 1)	**0.945**	**0.896**	0.889	0.743	**0.346**	0.570	**0.937**	0.792	0.641	0.851	0.877	0.766	0.736	**0.448**	0.780	0.717
		MSAN	(1, 0, 2)	0.890	0.848	**0.894**	**0.757**	0.335	0.559	0.935	**0.796**	0.636	0.847	0.872	0.769	0.743	0.440	**0.794**	0.718
	50%	Frequency	(0, 0, 0)	0.000	0.000	0.000	0.000	0.000	0.015	0.000	0.000	0.287	0.391	0.655	0.436	0.276	0.306	0.585	0.564
		SEER	(0, 1, 0)	0.000	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.288	0.396	0.651	0.435	0.276	0.302	0.567	0.571
		EAN	(0, 0, 1)	0.000	0.000	0.000	0.000	0.000	**0.530**	0.656	0.036	**0.562**	0.766	**0.885**	**0.785**	0.634	0.438	0.773	**0.734**
		REAP	(1, 1, 1)	0.006	0.006	0.000	0.000	0.000	0.333	0.849	0.690	0.506	0.710	0.865	0.700	0.617	0.382	**0.798**	0.701
		MoNE	-	0.000	0.000	0.000	0.000	0.000	0.454	0.632	0.016	0.555	**0.776**	**0.885**	**0.785**	**0.639**	**0.450**	0.776	**0.734**
		MAN	(1, 0, 1)	0.000	0.000	0.021	0.021	0.027	0.281	0.898	**0.792**	0.540	0.747	0.842	0.616	0.528	0.368	0.700	0.665
		MSAN	(1, 0, 2)	**0.256**	**0.250**	**0.370**	**0.328**	**0.060**	0.217	**0.910**	0.784	0.547	0.755	0.837	0.611	0.527	0.376	0.718	0.672

D Score Definitions in the Unified Formulation

Name	Formula	Hyperparameters (b, α, β)
Frequency	$\sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t]$	$(0, 0, 0)$
SEER	$\sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] g_{j,t}$	$(0, 1, 0)$
EAN	$\sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \lVert \mathbf{f}_{j,t} \rVert_2$	$(0, 0, 1)$
REAP	$\frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] g_{j,t} \lVert \mathbf{f}_{j,t} \rVert_2$	$(1, 1, 1)$
MoNE	$\left(\frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] g_{j,t} \right) \mathrm{Var}_j(\mathbf{f})$	-
MAN	$\frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \lVert \mathbf{f}_{j,t} \rVert_2$	$(1, 0, 1)$
MSAN	$\frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \lVert \mathbf{f}_{j,t} \rVert_2^2$	$(1, 0, 2)$
Gate-weighted EAN	$\sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] g_{j,t} \lVert \mathbf{f}_{j,t} \rVert_2$	$(0, 1, 1)$
Squared-gate activation energy	$\sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] g_{j,t}^2 \lVert \mathbf{f}_{j,t} \rVert_2^2$	$(0, 2, 2)$

Table 6: Score definitions for the scoring criteria in the unified formulation. Here $N_j = \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t]$. For MoNE, $\mathrm{Var}_j(\mathbf{f}) = \left\| \sqrt{\frac{1}{N_j - 1} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] (\mathbf{f}_{j,t} - \bar{\mathbf{f}}_j)^2} \right\|_2$, where $\bar{\mathbf{f}}_j = \frac{1}{N_j} \sum_{t=1}^{M} \mathbf{1}[j \in \mathcal{E}_t] \mathbf{f}_{j,t}$, and the square and square root are elementwise.