DIRCOV: DIAGNOSING EXPERT UNDER-ALLOCATION AND REDUNDANCY IN MIXTURE-OF-EXPERTS VIA DIRECTIONAL COVERAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) models activate a fixed top-k subset of experts per token, yet whether this single, layer-agnostic budget matches the coverage each layer actually needs has not been directly measured. We introduce *Directional Coverage* (DirCov), a lightweight, training-free diagnostic that quantifies how well the top-k expert combination reproduces the direction of the full mixture output, and *need*$_k$, the smallest budget a layer requires to reach 95% coverage. DirCov is defined on the data-side geometry of expert outputs under the model's own router. A parameter-space sanity check finds that OLMoE expert weights are near full-rank with near-zero mean pairwise cosine, ruling out simple weight-copy collapse as an explanation for the observed layer profiles. Applying DirCov to seven production MoE families spanning 16 to 128 experts, 16 to 48 layers, top-k from 2 to 8, differing normalisation, and both softmax and sparsemixer routing, we find three consistent phenomena: (i) directional coverage saturates monotonically with k; (ii) perplexity is U-shaped in inference k, so *over*-provisioning experts is actively harmful (up to a $555\times$ blow-up); and (iii) need_k is strongly heterogeneous across layers (a depth-driven, universal effect); across five text/math/code domains the per-layer profile is highly stable for most models ($\rho = 0.83$–0.95) but, in higher-capacity models, splits into a block structure along a natural-language$\leftrightarrow$structured boundary (within-group ρ up to 0.94 vs. across-group $\rho \approx 0.5$). Because a single global top-k cannot simultaneously satisfy under- and over-provisioned layers, we further show a bidirectional per-layer mismatch. The cross-architecture consistency indicates that fixed-k mismatch is a structural property of current MoE design rather than an artefact of any one model.

1 INTRODUCTION

Sparse Mixture-of-Experts (MoE) layers have become the dominant recipe for scaling large language models at fixed inference cost (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022; Dai et al., 2024). In every widely deployed MoE the same rule is applied at every layer: a router scores all E experts for each token and the top-k are activated, with k a single, hand-set, layer-agnostic constant (e.g. $k = 8$ for OLMoE and Qwen3-30B-A3B, $k = 2$ for Phi-3.5-MoE) (Muennighoff et al., 2024; Qwen Team, 2025; Abdin et al., 2024). This design implicitly assumes that a single budget is simultaneously appropriate for all layers and all inputs. To our knowledge this assumption has never been measured directly: existing analyses of MoE focus on *which* experts fire (routing entropy, load balance, expert utilisation) rather than on *how much of the layer's own computation the activated subset actually reconstructs*.

We ask a simpler, output-side question. Fix a layer and a token. Let y_{full} be the mixture output if *all* experts contributed under the router's own gate weights, and let y_k be the output when only the top-k are kept and re-normalised. How well does the direction of y_k cover y_{full}? We call this quantity *Directional Coverage*,

$$\mathrm{DirCov}(k) \;=\; \mathbb{E}_t\big[\cos^{+}\big(y_k(t),\, y_{\mathrm{full}}(t)\big)\big], \tag{1}$$

and define need_k as the smallest k for which $\mathrm{DirCov}(k) \geq 0.95$. DirCov is training-free, computed in a single forward pass, and depends only on the data-side geometry of expert outputs under the de-

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.

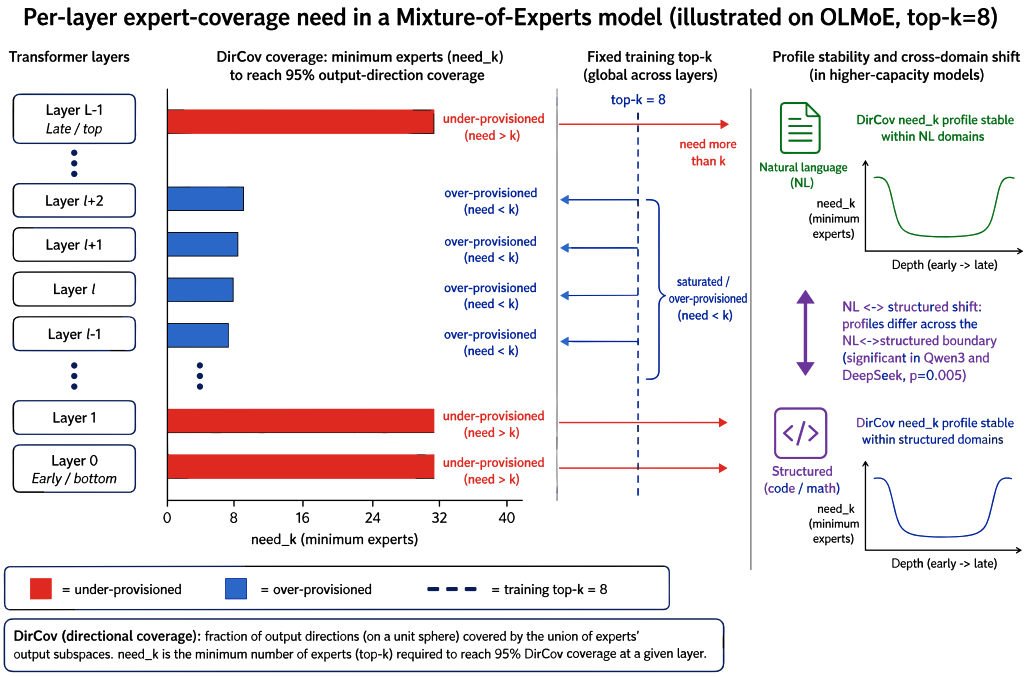

Figure 1: **Directional Coverage (DirCov) exposes a layer- and input-dependent activation budget that fixed top-k ignores.** For a layer and token, DirCov measures how well the direction of the top-k expert mixture y_k covers the full-mixture direction y_{full}; need_k is the smallest k reaching 95% coverage. Across seven production MoE families, need_k (i) saturates well before the expert pool, (ii) varies sharply with depth so a single global k_0 over-provisions some layers while under-provisioning others, and (iii) organises into a natural-language↔structured block structure in higher-capacity, fine-grained models.

ployed router—not on expert parameters, not on pairwise expert similarity, and not on any auxiliary loss.

DirCov turns the fixed-k assumption into a testable claim, and the test fails in three ways that are consistent across seven production MoE architectures (Fig. 1).

- **Monotone saturation.** $\text{DirCov}(k)$ rises monotonically with k and saturates well before E; most layers reach 95% coverage with far fewer experts than the full pool, but the training k_0 is not the saturation point.
- **Over-provisioning is harmful.** Sweeping the inference budget, perplexity is *U-shaped* in k: increasing k beyond a model-specific optimum degrades quality, with a $555\times$ perplexity blow-up for Phi-3.5-MoE at full activation. More experts are not free.
- **Heterogeneous, domain-stable need_k.** need_k varies sharply across layers ($\log_2$ variance up to 1.1)—a universal, depth-driven effect. Across five text/math/code domains the per-layer profile is highly stable for most models ($\rho = 0.83$–0.95), but in higher-capacity, fine-grained models it organises into a block structure along a natural-language↔structured boundary (within-group ρ up to 0.94 vs. across-group $\rho \approx 0.5$), significant in two independent architectures (p=0.005). The dominant mismatch is still *where* a layer sits; a secondary, capacity-dependent effect is *what* it reads.

Because early layers are systematically under-provisioned while mid-to-late layers are over-provisioned under the same global k_0, any single layer-agnostic budget *necessarily* mis-serves both ends—a bidirectional mismatch we quantify directly.

Our contribution is a diagnostic, not a new architecture. We (1) define DirCov and need_k, with a parameter-space sanity check that rules out simple expert-weight copy collapse in OLMoE; (2)

report the three phenomena above and show they replicate across 16–128 experts, 16–48 layers, top-k 2–8, two normalisation schemes, and softmax vs. sparsemixer routing; and (3) demonstrate the bidirectional per-layer mismatch that a single global k cannot avoid. The cross-architecture consistency is the central finding: fixed-k mismatch is structural, motivating layer-adaptive expert budgeting as a concrete open problem.

2 Related Work

From local experts to sparse transformers. Mixtures of local experts began with learned gating and hierarchical EM (Jacobs et al., 1991; Jordan & Jacobs, 1994), later extended to deep factored experts (Eigen et al., 2013). Sparsely gated layers (Shazeer et al., 2017) and transformer-scale systems (Lepikhin et al., 2020; Fedus et al., 2022; Zoph et al., 2022; Du et al., 2021; Artetxe et al., 2022) established token-wise conditional computation with a fixed top-k. BASE and hash layers explore balanced assignment or fixed routing (Lewis et al., 2021; Roller et al., 2021); sparse upcycling and MoEfication convert dense networks into experts (Komatsuzaki et al., 2022; Zhang et al., 2022). DirCov instead audits whether a deployed layer's fixed budget covers its own full mixture output.

Routing, balance, and specialisation. Prior work changes *which* experts fire through expert-choice routing, differentiable selection, stabilized routers, soft assignment, or null experts (Zhou et al., 2022; Hazimeh et al., 2021; Dai et al., 2022; Puigcerver et al., 2023; Zeng et al., 2024). Other studies ask whether learned routing or explicit objectives induce useful specialisation (Dikkala et al., 2023; Krishnamurthy et al., 2023; Shen et al., 2023; Nie et al., 2021). These input-side statistics and training interventions address balance, stability, or modularity. DirCov is output-side: it asks whether the selected experts reconstruct the layer's own full-mixture direction. A balanced router can therefore remain under- or over-provisioned.

Open MoE families and efficient execution. Current open families span shared/fine-grained experts, sparsemixer and softmax routers, and different expert granularities (Dai et al., 2024; DeepSeek-AI, 2024; Jiang et al., 2024; Xue et al., 2024; Muennighoff et al., 2024; Abdin et al., 2024; Qwen Team, 2024; 2025; ERNIE Team, Baidu, 2025). Systems such as MegaBlocks make these sparse models efficient to train (Gale et al., 2022). Across this design range, however, k remains a layer-agnostic constant; our seven-family evaluation tests the consequence of that shared assumption rather than proposing another router.

Geometry, theory, and budget adaptation. Parameter-side spectral geometry views MoE as soft clustering (Liu, 2026), while covering-number analyses study worst-case generalisation (Mayaki, 2026). Architectural alternatives alter the unit or sparsity of routing (Vejendla, 2025); DirCov instead measures data-side output coverage in an unchanged model. To our knowledge, none quantifies per-layer directional coverage, the over-provisioning U-shape, or the cross-architecture heterogeneity of the minimum sufficient budget need_k.

3 Directional Coverage

Setup. Consider one MoE layer with E experts $\{f_1, \dots, f_E\}$ and a router that, for token representation x_t, produces gate logits and a normalised gate distribution $g_t \in \Delta^{E-1}$ over the experts it would select. At inference the layer keeps a top-k subset $S_k(t) \subseteq \{1, \dots, E\}$ and re-normalises the gates over $S_k(t)$. We compare two outputs for the *same* token and layer:

$$y_{\text{full}}(t) \;=\; \sum_{e=1}^{E} \tilde{g}_{t,e}\, f_e(x_t), \qquad y_k(t) \;=\; \sum_{e \in S_k(t)} \frac{g_{t,e}}{\sum_{e' \in S_k(t)} g_{t,e'}}\, f_e(x_t), \tag{2}$$

where $\tilde{g}$ is the router's gate distribution over all E experts (i.e. the counterfactual "dense" mixture the same router would produce), and y_k is the actually-computed top-k output with gates re-normalised over the retained set. Both use the model's *own* router weights; no retraining or re-routing is performed.

Directional Coverage. We measure how well the top-k output covers the direction of the full mixture output:

$$\mathrm{DirCov}(k) \;=\; \mathbb{E}_t\big[\cos^+\big(y_k(t),\, y_{\mathrm{full}}(t)\big)\big]\,, \qquad \cos^+(a,b) = \max\!\left(0,\; \frac{\langle a,b\rangle}{\|a\|\,\|b\|}\right), \tag{3}$$

averaged over all tokens t in the evaluation set. We use a *directional* (cosine) rather than magnitude criterion because the downstream residual stream is dominated by direction after layer normalisation, and because y_k and y_{full} differ in norm by construction (fewer terms); $\cos^+$ isolates whether the retained experts point the layer the right way. $\mathrm{DirCov}(k) \in [0,1]$, is 1 when $k{=}E$, and is monotone non-decreasing in k in expectation.

Per-layer requirement need_k. For each layer ℓ we define the coverage-95 requirement

$$\mathrm{need}_k(\ell) \;=\; \min\big\{\, k : \mathrm{DirCov}_\ell(k) \geq 0.95 \,\big\}, \tag{4}$$

the smallest budget at which layer ℓ reproduces 95% of its own directional output. A model's deployed constant k_0 can then be compared to the profile $\{\mathrm{need}_k(\ell)\}_\ell$: layers with $\mathrm{need}_k(\ell) > k_0$ are *under-provisioned* (they lose coverage at the deployed budget), layers with $\mathrm{need}_k(\ell) < k_0$ are *over-provisioned* (they pay for experts they do not need). We summarise heterogeneity by the variance of $\log_2 \mathrm{need}_k(\ell)$ across layers.

Parameter-space non-collapse check. A trivial explanation for apparent coverage structure would be duplicated expert weights. We test this possibility in §4.1 using the cosine Gram matrix of each layer's flattened expert `down_proj` weights. This check is deliberately narrow: it can rule out low-rank weight-copy collapse, but not functional redundancy of expert outputs or degeneration of the router distribution.

4 Experiments

Models and data. We evaluate seven open, production MoE families that jointly span the design space: OLMoE-1B-7B (Muennighoff et al., 2024) (E=64, k_0=8, 16 layers), Qwen3-30B-A3B (Qwen Team, 2025) (E=128, k_0=8, 48 layers), Qwen1.5-MoE-A2.7B (Qwen Team, 2024) (E=60, k_0=4, 24 layers, with a shared expert), Phi-3.5-MoE (Abdin et al., 2024) (E=16, k_0=2, 32 layers, sparsemixer router), DeepSeek-MoE-16B (Dai et al., 2024) (E=64 fine-grained routed experts, k_0=6, 27 MoE layers, two shared experts), ERNIE-4.5-21B-A3B (ERNIE Team, Baidu, 2025) (E=64, k_0=6, 27 MoE layers, two shared experts, aux-loss-free routing), and DeepSeek-V2-Lite (DeepSeek-AI, 2024) (E=64 fine-grained routed experts, k_0=6, 26 MoE layers, softmax routing, no shared expert on the routed path). Unless noted, DirCov and perplexity are computed on WikiText (Merity et al., 2017); cross-domain analyses use five domains: two natural-language (WikiText, SQuAD v2 contexts (Rajpurkar et al., 2018)) and three structured (GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021)). All diagnostics are single-forward-pass, training-free, and use the models' own routers.

4.1 Expert weights do not exhibit copy collapse

Before reporting coverage phenomena, we test a narrow degeneracy hypothesis on OLMoE: the experts within a layer might be near-duplicates in parameter space. For each layer we form the cosine Gram matrix of flattened expert `down_proj` weights. Its normalised spectral effective rank lies in $[0.999980, 0.999984]$, while mean off-diagonal cosine lies within $[-2.7\times10^{-5}, 5.0\times10^{-5}]$. The expert-weight directions are therefore near full-rank and nearly orthogonal, ruling out simple weight-copy collapse. This sanity check does not rule out functional redundancy on token activations or degeneration of router probabilities, and we do not use it as evidence that DirCov is statistically independent of all similarity measures.

4.2 Monotone saturation and the perplexity U-shape

Figure 2 sweeps the inference budget and plots $\mathrm{DirCov}(k)$ against the active fraction k/E. Across all seven architectures DirCov rises monotonically and saturates well before E: OLMoE reaches

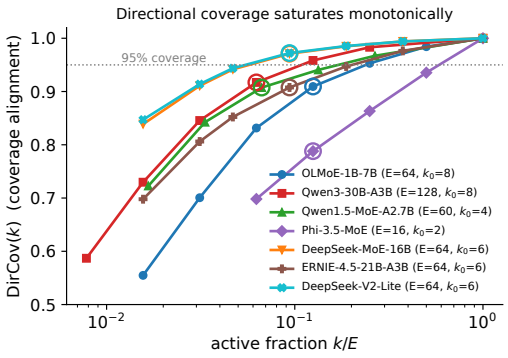

Figure 2: Directional coverage $\mathrm{DirCov}(k)$ vs. active fraction k/E (log-x). All seven architectures saturate monotonically; hollow circles mark the deployed k_0, which sits below the 95% line for most models.

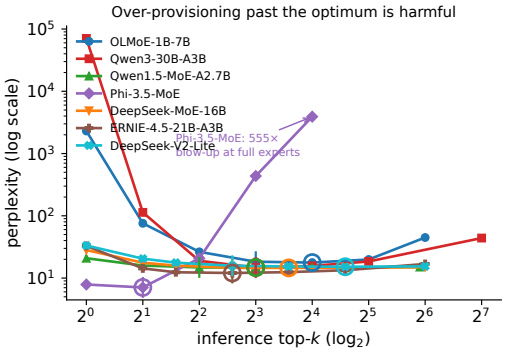

Figure 3: Perplexity vs. inference top-k (log–log). U-shaped for every model; over-provisioning is harmful, catastrophically so for Phi-3.5-MoE (555× blow-up). Hollow circle = best k; tick = deployed k_0.

0.909 at its training k_0=8 and 0.95 only around k=16; the Qwen models cross 0.92–0.94 at k_0=8 (k_0=4 for Qwen1.5). Crucially the training k_0 is *not* the coverage-95 point, foreshadowing per-layer mismatch.

Figure 3 sweeps perplexity against inference k. Contrary to the intuition that more experts can only help, perplexity is *U-shaped*: every model has an optimum near (but not at) k_0, and *over*-provisioning degrades quality. The effect is extreme for Phi-3.5-MoE, whose sparsemixer top-2 design makes full activation catastrophic—perplexity blows up from 7.07 at k=2 to 3913 at k=16, a 555× increase (Table 2). Even in the softmax-routed models, going from the optimum to full experts multiplies perplexity by 2–3×. The rank correlation between DirCov and perplexity is negative for the over-provisioning-tolerant models ($\rho \approx -0.5$ to -0.6) and strongly *positive* for Phi ($\rho = +0.9$), because for Phi more coverage co-occurs with the ruinous over-provisioning regime.

4.3 Heterogeneous per-layer need_k, and where domain stability holds

Figure 4 shows OLMoE's need_k per layer across WikiText, GSM8K, and code. The first, robust fact is that need_k is strongly *heterogeneous across layers*: early layers (0–7) demand need_k=32 while mid-to-late layers (10–14) are satisfied by need_k=8. This holds in every architecture ($\log_2 \mathrm{need}_k$ variance 0.19–1.10, Table 2); the coverage requirement is governed by a layer's *depth*.

Whether the *profile* is stable across data domains is more nuanced, and we report it honestly. We measure the per-layer need_k profile on *five* domains—two natural-language (WikiText, SQuAD v2 contexts) and three structured (GSM8K, HumanEval, MBPP)—and summarise cross-domain Spearman ρ grouped by whether a domain pair lies *within* the natural-language set, *within* the structured set, or *across* the two (Table 1). For OLMoE, Phi-3.5-MoE, and ERNIE-4.5-21B-A3B the profile is highly stable everywhere (ρ=0.83–0.95 in all three groups): the coverage requirement is essentially input-agnostic. For the higher-capacity, more finely partitioned models a clear *block* structure emerges: within-group correlations stay high (up to 0.94) while the across-group (natural-language ↔ structured) correlation drops sharply. A one-sided Mann–Whitney test of within- vs. across-group ρ confirms the block structure for *two independent* architectures—Qwen3-30B-A3B (0.78 vs. 0.55, p=0.005) and DeepSeek-MoE-16B (0.93 vs. 0.73, p=0.005)—and shows the same direction for Qwen1.5-MoE (0.70 vs. 0.51, p=0.13, limited by only 10 domain pairs), while OLMoE, Phi, and ERNIE show no such gap (p>0.11). The five-domain view thus sharpens the picture from a two-way split: *layer heterogeneity is universal and depth-driven*, and a *secondary domain effect* organises the profile along the natural-language↔structured boundary, growing more pronounced with model capacity. A fixed global k is therefore mismatched by depth for every model, and additionally by domain in the higher-capacity models whose layers specialise more sharply.

Table 1: Cross-domain stability of the per-layer need_k profile over *five* domains, as mean Spearman ρ grouped by pair type: within natural-language (WikiText, SQuAD), within structured (GSM8K, HumanEval, MBPP), and across the two. Heterogeneity across layers is universal; OLMoE/Phi are domain-stable in all groups, whereas the higher-capacity, fine-grained models (the larger Qwen models and DeepSeek-MoE) show a sharp within-vs-across gap—the profile shifts along the natural-language↔structured boundary.

Model	within-NL	within-struct	across (NL↔struct)
OLMoE-1B-7B	0.95	0.91	0.89
Phi-3.5-MoE	0.93	0.85	0.84
Qwen3-30B-A3B	0.65	0.82	0.54
Qwen1.5-MoE-A2.7B	0.90	0.63	0.51
DeepSeek-MoE-16B	0.90	0.94	0.73
ERNIE-4.5-21B-A3B	0.83	0.91	0.84
DeepSeek-V2-Lite	0.67	0.47	0.52

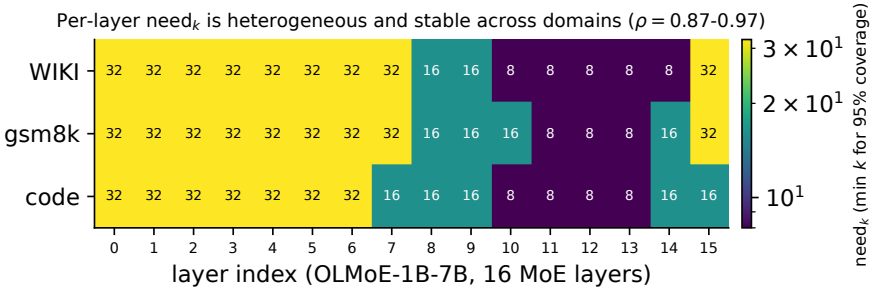

Figure 4: Per-layer need_k (min k for 95% coverage) for OLMoE across three representative domains (of five measured; Table 1). Heterogeneous across layers (early layers need ~ 32, mid-late need ~ 8); for OLMoE the profile is also highly domain-stable (ρ=0.87–0.97), whereas larger models show a natural-language↔structured shift.

4.4 A SINGLE GLOBAL k CANNOT FIT ALL LAYERS (BIDIRECTIONAL MISMATCH)

Given a deployed k_0, we classify each layer as under-provisioned ($\mathrm{need}_k > k_0$), over-provisioned ($\mathrm{need}_k < k_0$), or matched. For OLMoE at k_0=8, layers $\{0, 1, 2, 3, 4, 5\}$ are under-provisioned and layers $\{10, 11, 12, 13, 14\}$ are over-provisioned, with layer 9 borderline. Thus *no* single global k can satisfy both ends simultaneously: raising k to cover the early layers over-provisions the late layers (into the harmful regime of Fig. 3), and lowering it starves the early layers. The mismatch is bidirectional and unavoidable under a layer-agnostic budget.

The bidirectional nature of the mismatch has a direct consequence for tuning. Because raising k_0 moves the over-provisioned layers deeper into the harmful U-shaped regime of Fig. 3 while only partially relieving the starved early layers, a practitioner searching for the single best global k is forced onto a Pareto front where improving one group of layers necessarily degrades another. Fig. 5 makes this explicit for OLMoE: at the deployed k_0=8, roughly a third of the layers still demand more capacity while another third are already past their coverage plateau. This is not a tuning artefact—it is a structural limitation of using a single scalar to budget a network whose per-layer demand spans a $4\times$ range.

4.5 CROSS-ARCHITECTURE REPLICATION

Table 2 collects the headline numbers for all seven families. Despite spanning $E \in \{16, 60, 64, 128\}$, layers $\{16, 24, 32, 48\}$, $k_0 \in \{2, 4, 8\}$, two normalisation schemes, a shared-expert variant, and both softmax and sparsemixer routing, all three phenomena—monotone saturation, U-shaped perplexity with harmful over-provisioning, and heterogeneous need_k—appear in every model. This consistency is our central result: fixed-k mismatch is a *structural* property of current MoE design, not an artefact of any single architecture.

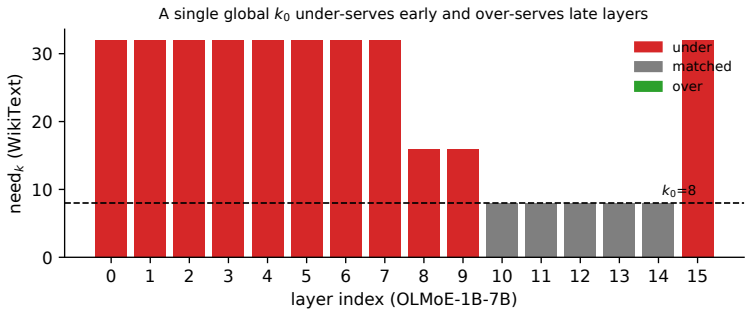

Figure 5: Per-layer provisioning for OLMoE at k_0=8. Each layer is classified by comparing its need_k to the deployed budget: *under*-provisioned ($\mathrm{need}_k > k_0$, early layers), *matched* ($\mathrm{need}_k = k_0$), or *over*-provisioned ($\mathrm{need}_k < k_0$, mid-to-late layers). The simultaneous presence of both under- and over-provisioned layers is precisely what makes any single global k a compromise: no horizontal line can meet the per-layer demand everywhere.

Table 2: Cross-architecture replication. k_0 = deployed top-k; ppl@k_0 / best ppl at deployed vs. optimal budget; *over-prov. blow-up* = ppl at full experts ÷ best ppl; mean $\mathrm{DirCov}(k_0)$ = per-layer coverage at deployed budget, averaged over layers with a 95% bootstrap CI (5,000 resamples over layers). All three phenomena hold for every model.

Model	E	layers	k_0	ppl@k_0	best ppl	over-prov. blow-up	mean $\mathrm{DirCov}(k_0)$ [95% CI]
OLMoE-1B-7B	64	16	8	18.30	17.73	2.5×	0.909 [0.891, 0.928]
Qwen3-30B-A3B	128	48	8	14.74	14.74	3.0×	0.917 [0.911, 0.923]
Qwen1.5-MoE-A2.7B	60	24	4	14.82	14.73	1.0×	0.907 [0.895, 0.919]
Phi-3.5-MoE	16	32	2	7.07	7.07	**555**×	0.788 [0.762, 0.814]
DeepSeek-MoE-16B	64	27	6	14.62	14.56	1.0×	0.971 [0.968, 0.975]
ERNIE-4.5-21B-A3B	64	27	6	12.05	12.05	1.4×	0.908 [0.891, 0.922]
DeepSeek-V2-Lite	64	26	6	15.79	15.18	1.0×	0.972 [0.969, 0.976]

4.6 Per-layer demand is heterogeneous across every architecture

The OLMoE mismatch generalises: every architecture has a non-flat depth profile, although its shape is model-specific (Fig. 8). Phi-3.5-MoE is under-provisioned throughout at k_0=2, whereas DeepSeek-MoE-16B matches most middle layers but over-serves final layers whose need_k falls to 1–3. Together with the domain structure in Table 1, this motivates the per-layer reallocation tested next.

4.7 A diagnosis–intervention gap: need_k-guided reallocation does not beat uniform

It is tempting to read the per-layer need_k profile as a prescription: simply give each layer $\mathrm{need}_k(\ell)$ experts. We test this on the two extremes of our suite—OLMoE (16 layers) and Qwen3-30B-A3B (48 layers). Because the summed per-layer demand $\sum_\ell \mathrm{need}_k(\ell)$ exceeds the deployed compute budget $B_0 = L\,k_0$ (for OLMoE, 360 vs. 128, a $+181\%$ overshoot—no layer is truly over-provisioned relative to k_0), a like-for-like comparison must fix the total budget B. We therefore compare, at several equal budgets B, a *uniform* allocation ($k_\ell = \lfloor B/L \rfloor$) against a need_k*-guided* allocation (a protect-then-boost rule that floors every layer at $\min(\mathrm{need}_k(\ell), \lfloor B/L \rfloor)$ and then spends the remaining budget on the layers with the largest unmet demand), plus a *random* equal-budget control. Figure 6 reports WikiText perplexity as a function of the average experts per layer B/L.

Three findings emerge, and they sharpen rather than soften the paper's message. *(i)* At *equal* budget, need_k-guided reallocation is statistically indistinguishable from uniform (OLMoE: 18.30 vs. 18.30 at B/L=8, 17.82 vs. 17.87 at B/L=10; Qwen3: 13.81 vs. 13.81, 14.05 vs. 14.02)—the need_k signal, though a faithful *diagnostic* of per-layer coverage, is *not* a sufficient control signal to out-allocate a flat budget. *(ii)* The random control is consistently worse (OLMoE 18.8–19.7, Qwen3 14.0–15.2), confirming that uniform is already a strong baseline and that naïve perturbations of the

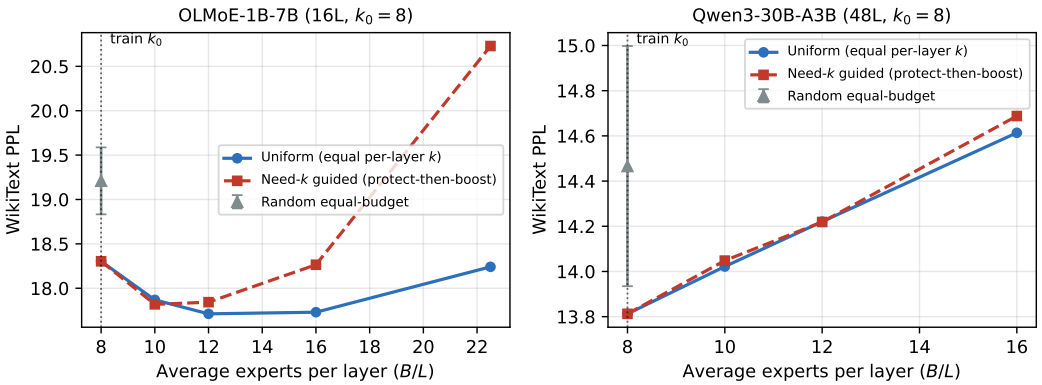

Figure 6: Per-layer budget reallocation at fixed total compute (WikiText PPL vs. average experts per layer B/L). At equal budget, *need*$_k$-guided protect-then-boost reallocation (red) tracks the *uniform* baseline (blue) essentially exactly, while a *random* equal-budget allocation (grey, mean±sd over seeds) is consistently worse. Increasing the budget past the deployed k_0 (dotted line) raises perplexity—a per-layer echo of the global U-shape. DirCov diagnoses the mismatch precisely, but a one-shot *need*$_k$ read-off does not translate into a better allocation: a *diagnosis–intervention gap* that frames adaptive per-layer budgeting as an open problem.

per-layer budget hurt. *(iii)* Raising the total budget above the deployed $L\,k_0$ *increases* perplexity for both models (OLMoE 18.24 at B/L=22.5, Qwen3 14.61 at B/L=16), a layer-wise echo of the global U-shape of Fig. 3: more experts inject residual-stream noise faster than they add useful coverage. Together these results expose a *diagnosis–intervention gap*: DirCov cleanly *identifies* which layers are under- and over-provisioned, but a direct, need$_k$-driven per-layer reallocation neither beats uniform at fixed compute nor tolerates a larger budget. This is not a weakness of the diagnostic but a statement about the problem: fixed-k mismatch is a genuine, *hard* problem for which the most obvious fix—spend the same compute where the diagnostic says it is needed—does not suffice, motivating future work on jointly learned or router-aware per-layer budgets rather than a one-shot read-off.

4.8 The inference-k trade-off transfers to task accuracy

The same non-monotonicity appears beyond perplexity. On GSM8K (200 examples, four-shot chain-of-thought, exact match), OLMoE peaks at its deployed k_0=8 with 40.0% accuracy, then falls to 23.0% at k=32 and 0.5% at full activation. Qwen3 peaks at 95.5% for k=8–16, then drops to 83.0% at k=64 and 64.8% at k=128 (the final point has n=125 completed examples). Thus harmful over-provisioning transfers to downstream exact-match accuracy rather than being an artifact of language-model perplexity (Fig. 7).

5 Discussion and Conclusion

We introduced Directional Coverage (DirCov), a training-free, single-pass diagnostic that asks whether an MoE layer's fixed top-k subset reproduces the direction of its own full mixture output, together with the per-layer requirement need$_k$. DirCov is output-side and data-dependent; a parameter-space sanity check additionally rules out simple expert-weight copy collapse in OLMoE, without claiming to exclude functional redundancy or router degeneracy. Across seven production MoE families spanning a wide design range we found three consistent phenomena—monotone coverage saturation, a perplexity U-shape in which over-provisioning is actively (and for Phi-3.5-MoE catastrophically, 555×) harmful, and heterogeneous per-layer need$_k$ (universally depth-driven, with domain stability that holds for most models and degrades along a natural-language↔structured shift in larger ones)—and showed that a single global k necessarily leaves some layers under-provisioned and others over-provisioned.

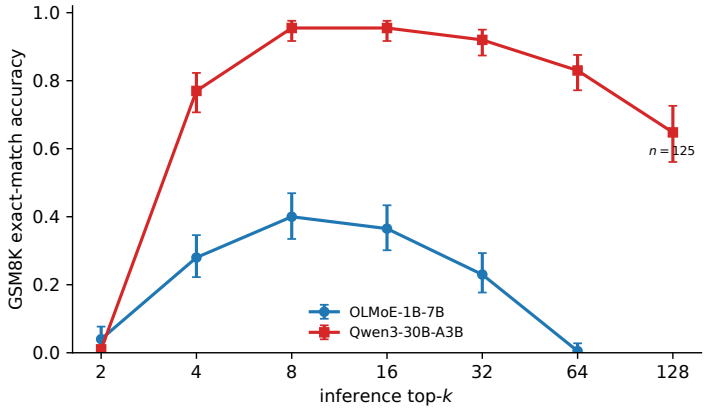

Figure 7: GSM8K exact-match accuracy under inference top-k sweeps, with Wilson 95% confidence intervals. Both models peak near their deployed budgets and degrade under over-provisioning. Each point uses n=200, except Qwen3 at k=128 (n=125 completed examples).

Implications. The cross-architecture consistency argues that fixed, layer-agnostic top-k is a structural mismatch rather than a per-model tuning gap, which motivates *layer-adaptive* expert budgeting as a concrete open problem. Our reallocation experiment (§4.7) sharpens what that problem is: a one-shot, need_k-guided read-off does *not* beat a uniform budget at fixed compute, and enlarging the budget only hurts. DirCov/need_k is thus best read as a faithful *criterion* for evaluating a layer-adaptive budget, not as a plug-in allocator—closing the diagnosis–intervention gap will likely require jointly learned or router-aware per-layer budgets. Because layer heterogeneity is universal and depth-driven, a depth-indexed budget profile is a natural first target; where the profile also proves domain-stable it can be estimated once and reused, while larger models may warrant a mild domain-conditioned adjustment. Because over-provisioning is harmful, matching (not merely upper-bounding) the budget matters.

Limitations. DirCov is a diagnostic, not an intervention: we quantify the mismatch and its harm, and we show that the obvious one-shot fix (a need_k-guided reallocation) does not beat uniform, but we do not propose a new architecture; adaptive-k inference also raises systems questions (variable compute per layer) we leave to future work. We report directional (cosine) coverage; magnitude effects on the residual stream are secondary but not zero. Our study covers seven open MoE families and five domains; closed models and further modalities remain to be checked, though the observed cross-architecture consistency is encouraging.

We release all diagnostic code and per-layer profiles to make the fixed-k mismatch reproducible and to seed work on layer-adaptive MoE budgeting.

ETHICS STATEMENT

This work is a purely diagnostic, analysis-only study of publicly released Mixture-of-Experts language models. We introduce no new model, deploy no system, and collect no human-subject or personally identifiable data. All experiments use open-weight checkpoints and standard public evaluation corpora (WikiText, SQuAD, GSM8K, HumanEval, MBPP) under their respective licenses. The diagnostic we propose (Directional Coverage) is training-free and read-only with respect to the underlying models, and does not modify or fine-tune any weights. We foresee no direct negative societal impact beyond those already inherent to the pre-trained models we study; if anything, our analysis is intended to make the internal expert-utilisation behaviour of such models more transparent and auditable. No sensitive or restricted data were used.

REPRODUCIBILITY STATEMENT

We have taken several steps to ensure our results are reproducible. The Directional Coverage and need$_k$ diagnostics are defined precisely in Section 3, and all evaluation settings (models, expert counts, layer counts, training top-k_0, the top-k sweep grid, evaluation domains, and the 95% coverage threshold) are stated in Section 4. We release the complete set of raw evaluation artifacts as JSON-lines files (`results/*.jsonl`) together with the scripts that regenerate every figure and table in both the main paper and this supplement (`code/make_figures.py`, `code/make_supp_figures.py`, `code/make_supp_tables.py`). Given the released artifacts and scripts, all quantitative claims (per-layer need$_k$ profiles, perplexity/DirCov sweeps, cross-domain Spearman correlations, provisioning classifications, and the expert-weight non-collapse sanity check) can be recomputed deterministically. The supplementary material further reports the full per-layer and per-domain tables underlying the aggregate statistics quoted in the main text.

USE OF LARGE LANGUAGE MODELS

Large language models were used only as general-purpose writing and coding assistants: to help polish prose for clarity and grammar, and to assist with routine plotting and table-formatting code. They were not used to generate research ideas, to design or run the experiments, to analyse or interpret results, or to produce any of the reported quantitative findings. All experimental design, data analysis, and scientific claims are the authors' own, and the authors take full responsibility for the content of the paper.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11699–11732, 2022. doi: 10.18653/v1/2022.emnlp-main.804.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7085–7095, 2022. doi: 10.18653/v1/2022.acl-long.489.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

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

Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9376–9396, 2023. doi: 10.18653/v1/2023.emnlp-main.583.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*, 2021.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

ERNIE Team, Baidu. ERNIE 4.5 technical report. `https://ernie.baidu.com`, 2025. Baidu ERNIE 4.5 model family technical report.

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

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. MegaBlocks: Efficient sparse training with mixture-of-experts. *arXiv preprint arXiv:2211.15841*, 2022.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. DSelect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *arXiv preprint arXiv:2106.03760*, 2021.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*, 2022.

Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. Improving expert specialization in mixture of experts. *arXiv preprint arXiv:2302.14703*, 2023.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. *arXiv preprint arXiv:2103.16716*, 2021.

Feilong Liu. Mixture-of-experts as soft clustering: Dual jacobian-pca spectral geometry. *arXiv preprint arXiv:2601.11616*, 2026.

Mansour Zoubeirou a Mayaki. Generalization and scaling laws for mixture-of-experts transformers. *arXiv preprint arXiv:2604.09175*, 2026.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, et al. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Xiaonan Nie, Xupeng Miao, Shijie Cao, Lingxiao Ma, Qibin Liu, Jilong Xue, Youshan Miao, Yi Liu, Zhi Yang, and Bin Cui. EvoMoE: An evolutional mixture-of-experts training framework via dense-to-sparse gate. *arXiv preprint arXiv:2112.14397*, 2021.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 784–789, 2018. doi: 10.18653/v1/P18-2124.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. *arXiv preprint arXiv:2106.04426*, 2021.

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

Yikang Shen, Zheyu Zhang, Tianyou Cao, Shawn Tan, Zhenfang Chen, and Chuang Gan. ModuleFormer: Modularity emerges from mixture-of-experts. *arXiv preprint arXiv:2306.04640*, 2023.

Harshil Vejendla. SliceMoE: Routing embedding slices instead of tokens for fine-grained and balanced expert specialization. *arXiv preprint arXiv:2510.04286*, 2025.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. OpenMoE: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. AdaMoE: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6223–6235, 2024. doi: 10.18653/v1/2024.findings-emnlp.361.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 877–890, 2022. doi: 10.18653/v1/2022.findings-acl.71.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, 2022. doi: 10.52202/068431-0515.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A Additional Figures and Tables

This appendix expands the aggregate results in the main paper into per-layer, per-domain, and per-budget views. The figures first establish where coverage demand varies, then test for simple expert-weight copy collapse and expose the resulting provisioning mismatch. The tables provide the corresponding numerical values so that the cross-model claims can be inspected without relying on visual summaries.

A.1 Coverage demand across depth and domains

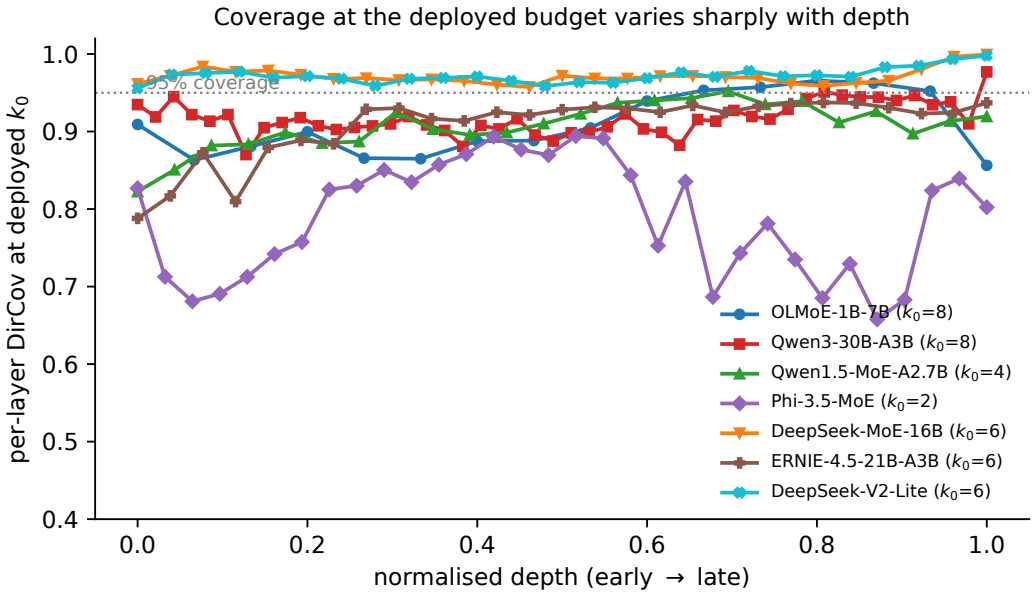

Figure 8: Per-layer Directional Coverage evaluated at each model's training top-k_0, plotted against normalised depth for all seven MoE families. Coverage is far from uniform across depth: early and late layers frequently fall below the mid-network plateau, indicating that a single global k_0 over- or under-serves different layers.

We begin with the two axes along which a global routing budget can fail: layer depth and input domain. Figure 8 shows the coverage delivered by each model's deployed k_0; Figures 9 and 10 convert it into the minimum sufficient budget and its dispersion.

These views support two distinct conclusions. First, heterogeneity across depth is universal: every architecture contains layers with different coverage requirements. Second, domain sensitivity is model-dependent. Some models preserve nearly the same depth profile across all five domains, whereas others exhibit the natural-language versus structured-domain block pattern quantified in Figure 14 and Table 10. Thus the primary failure of fixed k is depth mismatch; domain shift is an additional source of mismatch for a subset of architectures.

A.2 From coverage to provisioning mismatch

The next figures test alternative explanations and connect the diagnostic to the deployed routing decision. Figure 11 shows why average coverage alone cannot select k: increasing coverage eventually enters an over-provisioned regime in which perplexity worsens.

The parameter-space check in Figure 12 rules out the narrow explanation that nearly identical expert weights generate the observed pattern. It does not establish independence between DirCov and all possible output-similarity measures. The provisioning counts in Figure 13 then expose the operational consequence: at the same deployed k_0, under-provisioned and over-provisioned layers coexist.

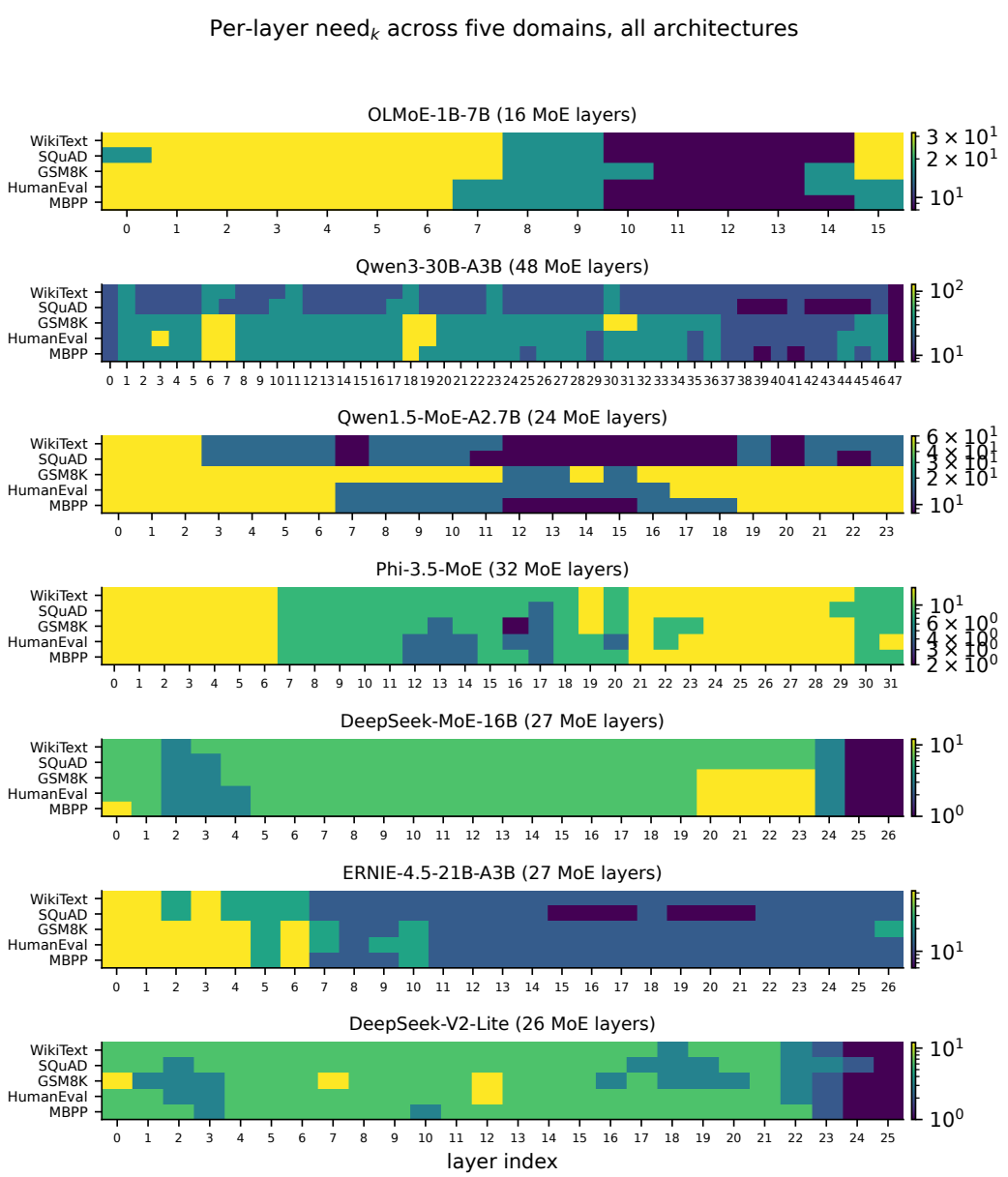

Figure 9: Heatmap of per-layer need$_k$ (the smallest k reaching 95% directional coverage) for every model $\times$ domain combination. Structural domains (code, MBPP, GSM8K) systematically demand different depths than natural-language domains (WikiText, SQuAD), and the demand pattern is model-specific.

This coexistence explains why globally increasing or decreasing k cannot resolve the mismatch and motivates the diagnosis–intervention gap studied in the main paper.

The correlation grid also shows why the domain effect should not be reduced to a single cross-model average. Uniformly high matrices indicate a stable depth skeleton, while block-structured matrices identify models whose coverage demand changes between natural-language and structured inputs. The grouped statistics in Table 10 retain this per-model distinction.

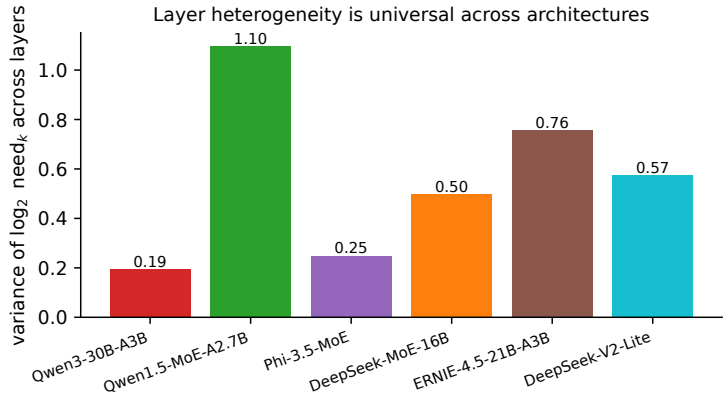

Figure 10: Variance of $\log_2 \text{need}_k$ aggregated over all layers and domains. Higher bars (Qwen1.5-MoE) indicate strongly heterogeneous depth demand, whereas lower bars (Phi-3.5-MoE) indicate a comparatively uniform requirement.

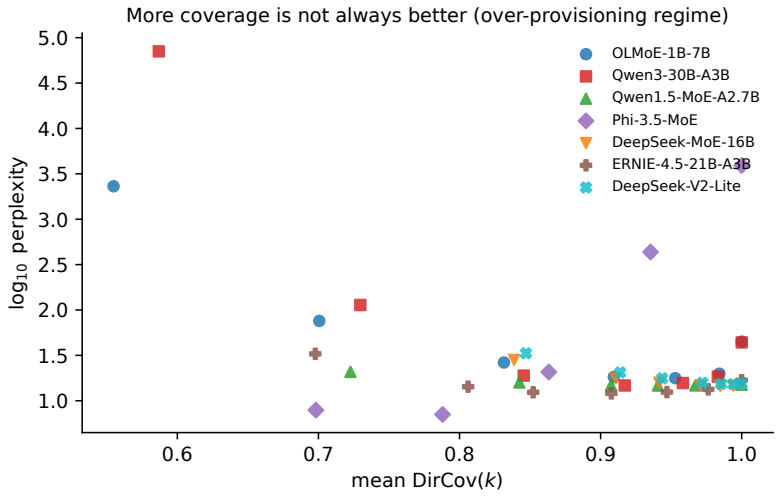

Figure 11: Sequence-averaged DirCov versus log perplexity across the top-k sweep for all models. The relationship is non-monotone in the small-k regime (where degenerate routing inflates coverage without improving loss), consistent with the negative rank correlations reported in the main paper.

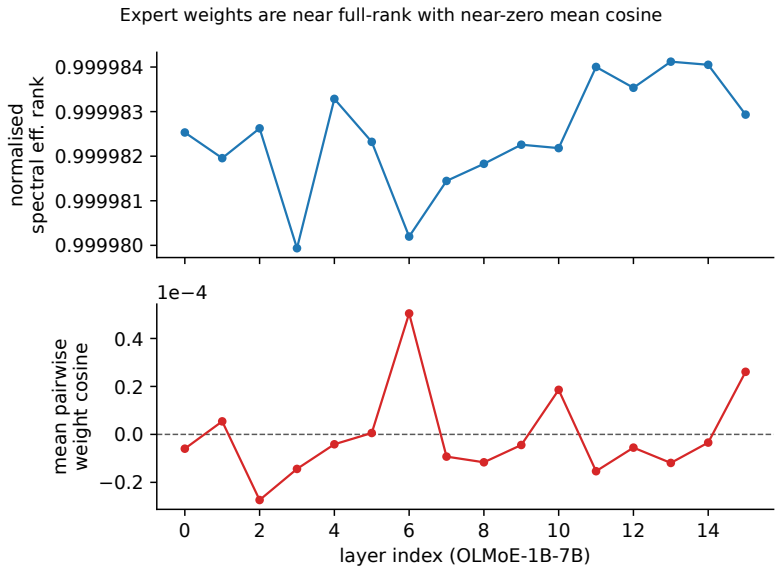

Figure 12: Parameter-space expert-weight non-collapse check for OLMoE. The top panel reports the normalised spectral effective rank of each layer's expert-weight cosine Gram matrix (1 denotes full rank); the bottom reports mean pairwise weight cosine (0 denotes orthogonality on average). Values near 1 and 0, respectively, rule out simple duplication of `down_proj` weights, but do not test functional output redundancy or router degeneration.

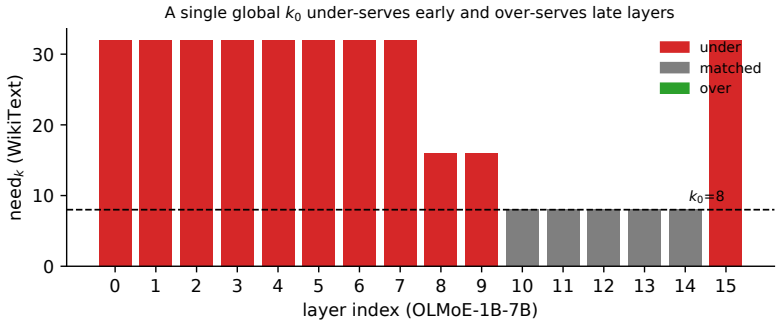

Figure 13: OLMoE per-layer provisioning classification at $k_0 = 8$: layers whose need$_k$ exceeds k_0 (under-provisioned) coexist with layers whose need$_k$ falls below k_0 (over-provisioned), demonstrating the bidirectional mismatch described in the main paper.

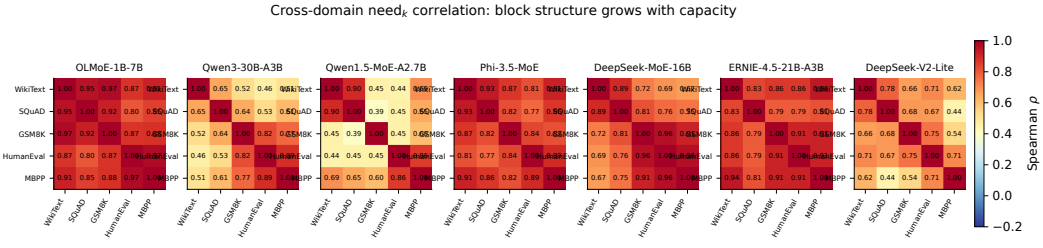

Figure 14: Per-model 5×5 pairwise Spearman ρ between per-layer need$_k$ profiles across the five evaluation domains. Within natural-language pairs correlation is high; cross-family (NL vs. structural) pairs are markedly weaker, quantifying domain-dependent depth demand.

Table 3: Per-model summary statistics: number of experts E, layer count, training top-k_0, median per-layer need$_k$ (WikiText), variance of $\log_2$ need$_k$ over all layers/domains, and the set of distinct need$_k$ values observed.

Model	E	layers	k_0	median need$_k$	$\log_2$need$_k$ var	need$_k$ values
OLMoE-1B-7B	64	16	8	32	0.724	8, 16, 32
Qwen3-30B-A3B	128	48	8	16	0.710	8, 16, 32, 128
Qwen1.5-MoE-A2.7B	60	24	4	16	1.394	8, 16, 60
Phi-3.5-MoE	16	32	2	16	0.441	2, 4, 8, 16
DeepSeek-MoE-16B	64	27	6	6	0.664	1, 3, 6, 12
ERNIE-4.5-21B-A3B	64	27	6	12	0.946	6, 12, 24, 64
DeepSeek-V2-Lite	64	26	6	6	0.614	1, 2, 3, 6, 12

Table 4: Per-model, per-domain, per-layer need$_k$ profiles across the five evaluation domains (natural language: WikiText, SQuAD; structural: GSM8K, HumanEval, MBPP), aggregated in Tables 4–8. Cross-domain rank agreement is summarised in Table 10.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WikiText	32	32	32	32	32	32	32	32	16	16	8	8	8	8	8	32
SQuAD	16	32	32	32	32	32	32	32	16	16	8	8	8	8	8	32
GSM8K	32	32	32	32	32	32	32	32	16	16	16	8	8	8	16	32
HumanEval	32	32	32	32	32	32	32	16	16	16	8	8	8	8	16	16
MBPP	32	32	32	32	32	32	32	16	16	16	8	8	8	8	8	16

B Detailed Tables

This section exposes the numerical substrate of the aggregate plots. We report the model-level summary first, followed by full per-layer demand profiles, cross-domain agreement, budget sweeps, provisioning counts, and expert-weight non-collapse statistics. These tables support claim verification and reveal architecture-specific exceptions hidden by pooled summaries.

B.1 Per-layer coverage requirements

Tables 4–9 provide the unaggregated evidence behind the depth-heterogeneity claim. They can be read row-wise to compare domains at a fixed layer and column-wise to trace a domain through depth. Repeated transitions among the candidate budgets show that the result is not driven by one anomalous layer. The degree of cross-domain agreement nevertheless differs by architecture, which is why we retain the per-model profiles rather than reporting only a pooled variance.

B.2 Cross-domain agreement and budget sweeps

Tables 11 and 12 should be interpreted jointly. Directional coverage rises toward saturation, while perplexity can turn upward after its optimum; maximum coverage is therefore not a valid stand-alone routing objective. Table 10 complements these sweeps by showing when the layer-wise demand ordering is stable across domains and when it changes by domain family.

B.3 Provisioning and parameter-space non-collapse check

Finally, Table 13 translates continuous coverage into the decision categories used in the paper: under-provisioned, matched, and over-provisioned relative to the deployed budget. Table 14 checks only that these categories are not trivially explained by duplicated expert `down_proj` weights. Near-unit normalised effective rank means near-full-rank parameter geometry, not rank one; near-zero mean cosine means that expert weights are orthogonal on average. This check does not exclude activation-level functional redundancy or router degeneration. Together the tables delimit the claim: DirCov diagnoses layer-specific mismatch in the observed models, but does not imply that a naive redistribution of the same global budget must improve perplexity.

Table 5: Per-layer need$_k$ profiles for Qwen3-30B-A3B across domains.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
WikiText	16	32	16	16	16	16	32	32	16	16	16	32	16	16	16	16	16	16	32	16	16	16	16	32	16	16	16	16	16	16	32	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	8
SQuAD	16	32	16	16	16	16	32	16	16	16	32	32	16	16	16	16	16	32	32	16	16	16	16	32	16	16	16	16	16	16	32	16	16	16	16	16	16	16	8	8	8	16	8	8	8	8	16	8
GSM8K	16	32	32	32	32	32	128	128	32	32	32	32	32	32	32	32	32	32	128	128	32	32	32	32	32	32	32	32	32	128	128	32	32	32	32	32	16	16	16	16	16	16	16	16	16	32	32	8
HumanEval	16	32	32	128	32	32	128	128	32	32	32	32	32	32	32	32	32	32	128	128	32	32	32	32	32	32	32	32	16	32	32	32	32	32	32	16	32	16	16	16	16	16	16	16	32	32	32	8
MBPP	16	32	32	32	32	32	128	128	32	32	32	32	32	32	32	32	32	32	128	32	32	32	32	32	32	16	32	32	32	32	32	32	32	32	32	16	32	16	16	8	16	8	16	16	32	16	32	8

Table 6: Per-layer need$_k$ profiles for Qwen1.5-MoE-A2.7B across domains.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
WikiText	60	60	60	16	16	16	16	8	16	16	16	16	8	8	8	8	8	8	8	8	16	8	16	16
SQuAD	60	60	60	16	16	16	16	8	16	16	16	8	8	8	8	8	8	8	8	16	8	16	8	16
GSM8K	60	60	60	60	60	60	60	60	60	60	60	60	16	16	60	16	60	60	60	60	60	60	60	60
HumanEval	60	60	60	60	60	60	60	16	16	16	16	16	16	16	16	16	16	60	60	60	60	60	60	60
MBPP	60	60	60	60	60	60	60	16	16	16	16	16	8	8	8	8	16	16	16	60	60	60	60	60

Table 7: Per-layer need$_k$ profiles for Phi-3.5-MoE across domains.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WikiText	16	16	16	16	16	16	16	8	8	8	8	8	8	8	8	8	8	8	8	16	8	16	16	16	16	16	16	16	16	16	8	8
SQuAD	16	16	16	16	16	16	16	8	8	8	8	8	8	8	8	8	4	8	16	8	16	16	16	16	16	16	16	16	16	8	8	8
GSM8K	16	16	16	16	16	16	16	8	8	8	8	8	4	8	8	2	4	8	16	8	16	8	8	16	16	16	16	16	16	16	8	8
HumanEval	16	16	16	16	16	16	16	8	8	8	8	8	4	4	4	8	4	4	8	8	4	16	8	16	16	16	16	16	16	16	8	16
MBPP	16	16	16	16	16	16	16	8	8	8	8	8	4	4	4	8	8	4	8	8	8	16	16	16	16	16	16	16	16	16	8	8

Table 8: Per-layer need$_k$ profiles for DeepSeek-MoE-16B across domains.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
WikiText	6	6	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	3	1	1
SQuAD	6	6	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	3	1	1
GSM8K	6	6	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	12	12	12	12	3	1	1
HumanEval	6	6	3	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	12	12	12	12	3	1	1
MBPP	12	6	3	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	12	12	12	12	3	1	1

Table 9: Per-layer need$_k$ profiles for ERNIE-4.5-21B-A3B across domains.

domain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
WikiText	64	64	24	64	24	24	24	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
SQuAD	64	64	24	64	24	24	24	12	12	12	12	12	12	12	12	6	6	6	12	6	6	6	12	12	12	12	12
GSM8K	64	64	64	64	64	24	64	24	12	12	24	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	24
HumanEval	64	64	64	64	64	24	64	24	12	24	24	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
MBPP	64	64	64	64	64	24	64	12	12	12	24	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12

Table 10: Mean pairwise Spearman ρ between per-layer need$_k$ profiles, decomposed into within-natural-language, within-structural, and across-family domain pairs, with the number of pairs in each group.

Model	within-NL	within-struct	across	n_{NL}	n_{st}	n_{ac}
OLMoE-1B-7B	0.95	0.91	0.89	1	3	6
Qwen3-30B-A3B	0.65	0.82	0.54	1	3	6
Qwen1.5-MoE-A2.7B	0.90	0.63	0.51	1	3	6
Phi-3.5-MoE	0.93	0.85	0.84	1	3	6
DeepSeek-MoE-16B	0.89	0.94	0.73	1	3	6
ERNIE-4.5-21B-A3B	0.83	0.91	0.84	1	3	6
DeepSeek-V2-Lite	0.78	0.66	0.63	1	3	6

Table 11: Perplexity as a function of evaluation top-k for all seven models ("–" denotes k values not in that model's sweep). The non-monotone, U-shaped behaviour is visible for several architectures.

Model	k=1	k=2	k=3	k=4	k=6	k=8	k=12	k=16	k=24	k=32	k=60	k=64	k=128
OLMoE-1B-7B	2305.34	75.76	–	26.38	–	18.30	–	17.73	–	19.87	–	44.79	–
Qwen3-30B-A3B	70641	113.52	–	18.92	–	14.74	–	15.67	–	18.48	–	–	43.91
Qwen1.5-MoE-A2.7B	20.81	15.90	–	14.82	–	14.73	–	14.75	–	–	15.02	–	–
Phi-3.5-MoE	7.88	7.07	–	20.71	–	435.56	–	3912.86	–	–	–	–	–
DeepSeek-MoE-16B	28.17	17.64	15.83	–	14.62	–	14.56	–	14.66	–	–	14.97	–
ERNIE-4.5-21B-A3B	32.87	14.30	12.39	–	12.05	–	12.45	–	13.35	–	–	16.83	–
DeepSeek-V2-Lite	33.33	20.50	17.66	–	15.79	–	15.35	–	15.18	–	–	15.42	–

Table 12: Sequence-averaged Directional Coverage as a function of evaluation top-k for all seven models.

Model	k=1	k=2	k=3	k=4	k=6	k=8	k=12	k=16	k=24	k=32	k=60	k=64	k=128
OLMoE-1B-7B	0.555	0.701	–	0.832	–	0.909	–	0.953	–	0.984	–	1.000	–
Qwen3-30B-A3B	0.587	0.730	–	0.846	–	0.917	–	0.958	–	0.983	–	–	1.000
Qwen1.5-MoE-A2.7B	0.723	0.842	–	0.907	–	0.941	–	0.967	–	–	1.000	–	–
Phi-3.5-MoE	0.698	0.788	–	0.863	–	0.935	–	1.000	–	–	–	–	–
DeepSeek-MoE-16B	0.839	0.911	0.942	–	0.971	–	0.985	–	0.994	–	–	1.000	–
ERNIE-4.5-21B-A3B	0.698	0.806	0.852	–	0.908	–	0.947	–	0.976	–	–	1.000	–
DeepSeek-V2-Lite	0.847	0.914	0.944	–	0.972	–	0.985	–	0.994	–	–	1.000	–

Table 13: Provisioning classification at each model's training k_0: number of under-provisioned (need$_k > k_0$), matched ($= k_0$), and over-provisioned ($< k_0$) layers on WikiText.

Model	k_0	#under	#matched	#over	#layers
OLMoE-1B-7B	8	11	5	0	16
Qwen3-30B-A3B	8	47	1	0	48
Qwen1.5-MoE-A2.7B	4	24	0	0	24
Phi-3.5-MoE	2	32	0	0	32
DeepSeek-MoE-16B	6	0	23	4	27
ERNIE-4.5-21B-A3B	6	27	0	0	27
DeepSeek-V2-Lite	6	0	21	5	26

Table 14: OLMoE parameter-space non-collapse summary. Normalised spectral effective rank is compared with its full-rank reference of 1; mean pairwise expert-weight cosine is compared with its orthogonality reference of 0. The ranges aggregate all 16 MoE layers.

Metric	Range across layers	Non-collapse reference
Normalised spectral effective rank	[0.999980, 0.999984]	1 (full rank)
Mean pairwise weight cosine	$[-2.74 \times 10^{-5}, 5.04 \times 10^{-5}]$	0 (orthogonal on average)