Router Prior Bias: Preserving Base Routing Structure in MoE Post-Training

Jaedeok Lee[1,2], Keonwoo Kim[1], Dongyoon Han[3], Sangdoo Yun[3], Yera Choi[1,†], Haanju Yoo[1,†]

[1]NAVER Applied AI Group
[2]Healthcare AI Research Institute (HARI), Seoul National University Hospital
[3]NAVER AI Lab

[†]Corresponding authors. **Correspondence:** yera.choi@navercorp.com, haanju.yoo@navercorp.com

Abstract

Mixture-of-Experts (MoE) pretraining relies on an auxiliary load-balancing loss (LBL) to drive per-expert utilization toward uniformity. Post-training inherits a different situation: the base router already encodes non-uniform expert co-activation structure, which a re-imposed uniformity objective flattens away. We show that downstream performance depends instead on holding this inherited routing *softly*, a principle we term soft router anchoring, and instantiate it as **Router Prior Bias (RPB)**, a training-time bias that pulls the router logits toward a prior read off the frozen base router while leaving the router itself trainable. On math post-training of Moonlight-16B-A3B, RPB attains 45.77 in-domain accuracy against 31.91 under re-applied LBL and 29.44 under unanchored fine-tuning, and retains more out-of-domain capability than either. The ordering against LBL reproduces on a second model family (Qwen3-30B-A3B-Base), and the advantage over LBL is resolvable on an independently sourced corpus. Anchors defined on the router weights, on its logits, or on its output distribution perform comparably with no consistent ordering, which places the effect in the softness of the constraint rather than in the particular prior RPB supplies. Retained community structure in the expert co-activation graph tracks these gains wherever the base router is non-uniform enough for communities to form, yet enforcing the same prior as a hard assignment preserves that structure while performance falls sharply. Community structure is therefore a footprint of soft anchoring rather than its source, and the practical lesson is that inherited routing should be held softly during post-training, since both flattening it toward uniformity and enforcing it absolutely carry a downstream cost. Our code will be released at github.com/naver-ai/rpb.

1 Introduction

Mixture-of-Experts (MoE) language models, including Qwen (Qwen Team, 2025), Mixtral (Jiang et al., 2024), DeepSeek (DeepSeek-AI, 2024a), and Moonlight (Liu et al., 2025), route each token to a small subset of experts, allowing total parameter count to grow without a commensurate increase in per-token computation (Cai et al., 2024). Because an unbalanced router would leave much of that capacity idle, MoE pretraining adds an auxiliary load-balancing loss (LBL) that penalizes deviations from uniform per-expert utilization (Lepikhin et al., 2020; Fedus et al., 2021).

By the time post-training begins, however, the router is no longer an unstructured allocator: pretraining has already induced non-uniform specialization and recurring patterns of expert co-activation (Tang et al., 2026; Lo et al., 2024), so a re-imposed uniformity objective works against structure the model has spent its pretraining budget acquiring. Academic adaptation methods and practitioner toolchains have accordingly converged on preserving that inherited structure rather than re-imposing pretraining-style uniformity: structure-aware adaptation methods select, retain, or route through it (Wang et al., 2024b; Li et al., 2025; Eo et al., 2025), while post-training toolchains drop load balancing altogether, with NVIDIA Megatron-Bridge exposing no-load-balancing and auxiliary-loss-free configurations (NVIDIA, 2025) and Unsloth disabling router-layer training entirely (Unsloth AI, 2025). These defaults are widely adopted, yet which property of the base router they preserve, and why that preservation benefits adaptation, remain underexamined.

One reason the question has stayed open is that the available defaults sit at the two extremes, either leaving base routing unconstrained or enforcing it in full. Router-freeze, the strictest of them, obtains routing stability by removing trainability altogether, which caps how far the model can adapt to the post-training corpus and answers two distinct questions at once: which routing structure to preserve, and how strictly to enforce it. Separating the two re-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 50281–50301
October 24-29, 2026 ©2026 Association for Computational Linguistics

quires an intervention whose enforcement strength is a continuous quantity rather than an architectural commitment. We therefore introduce **Router Prior Bias (RPB)**, a training-time soft logit bias that anchors routing to a prior formed from the frozen base router while leaving the router itself trainable.

To identify what such anchoring actually preserves, we measure routing retention at three levels: (1) *per-expert utilization*, the level classical LBL targets; (2) *token-level top-k selection*, which experts a given token is routed through; and (3) *expert co-activation community structure*, which experts tend to be selected together (Figure 1). Methods in the *soft router anchoring* family, RPB among them, reach high downstream performance while retaining base top-k selection and community structure, whereas re-applied LBL preserves per-expert utilization alone. Objectives that constrain the router parameters directly, without constructing any prior at all, behave the same way, so what the family shares is the anchoring rather than the prior RPB supplies.

This leaves open whether the gain comes from soft enforcement itself or from the community structure that anchoring happens to preserve, and we separate the two interventionally. Enforcing the same prior as a hard assignment preserves community structure at or above the level soft anchoring reaches, yet substantially reduces performance (Appendix A.1). Shuffling the prior then varies its content at fixed enforcement: a within-community shuffle applied softly leaves performance essentially intact, while the same prior applied as a hard constraint does not (§5.2.2). Preserved community structure is therefore a footprint of soft enforcement at training time rather than the source of the performance gain.

Our contributions are as follows:

1. **A controlled comparison of post-training routing objectives.** Re-applying the pre-training load-balancing loss during MoE post-training degrades downstream performance relative to leaving the inherited routing softly anchored, and the effect persists under a change of model family (Qwen3-30B-A3B-Base) and of corpus (OpenR1-Math-220k). Whether soft anchoring additionally outperforms an unanchored no-load-balancing baseline is model-dependent, and we report both directions.

2. **Soft router anchoring, and a method that implements it.** On Moonlight-16B-A3B, four objectives acting in parameter, logit, and probability space perform comparably with no consistent ordering, which locates the effect in the anchoring itself rather than in any one objective and makes it available to any method that holds the router near its base routing while leaving it trainable. RPB is the sample-conditional instance we develop and evaluate.

3. **A retention protocol and a diagnostic.** Measuring retention at three levels, with the training-time prior removed at evaluation, identifies community NMI as the level that separates soft anchoring from re-applied LBL, *in base models whose routing is non-uniform enough to form communities*, a condition Moonlight-16B-A3B satisfies and DeepSeek-V2-Lite does not.

4. **Cause separated from footprint.** Interventional controls that vary enforcement strength and prior content independently isolate soft enforcement, rather than community-structure preservation, as the source of the performance gain.

2 Preliminaries

We now make precise the three levels at which we measure *routing retention*, the degree to which a post-trained router still routes as its base router did: (1) per-expert utilization, (2) token-level top-k selection, and (3) expert co-activation community structure, all reported alongside downstream **performance** (in-domain and out-of-domain means). Per-expert utilization is a sanity condition rather than a result, since every method we compare falls in the same range on it. All three levels are measured *bias-free*: the training-time RPB prior-bias hook is removed at evaluation, so the community structure and top-k overlap we measure reflect the trained router rather than an inference-time prior.

Per-expert utilization. This is the level classical LBL targets. Between the post-training and base routers' expert distributions we report total variation distance (TVD) for L_1 marginal shift and Jensen–Shannon divergence (JSD) for symmetric, bounded shift.

Token-level top-k selection. For per-token selection identity we report **top-k overlap**: for each token, the intersection size between the base-router

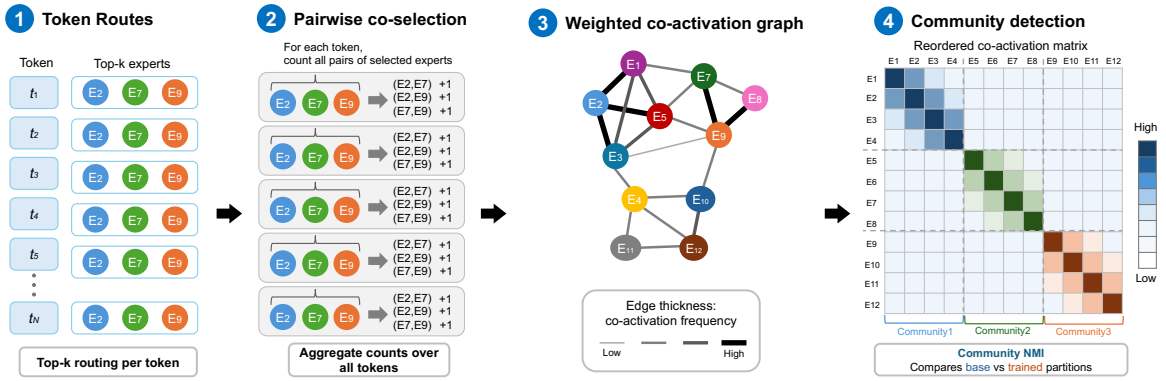


Figure 1: Construction of the layer-wise expert co-activation graph, in four steps. For each input token, the router selects its top-k experts, where E_i denotes expert i. Experts selected together for the same token are counted as co-activations. These co-activation counts are accumulated into a layer-wise weighted expert graph, with experts as nodes and co-activation frequency as edge weight. Community detection then partitions the graph into expert communities, and community NMI compares the base and post-trained partitions.

and post-training top-k sets, normalized by k and averaged over tokens and layers.

Expert co-activation community structure. For each transformer layer ℓ, we build an expert co-activation graph (Figure 1) whose nodes are experts and whose edge weight $w_{ij}^{(\ell)}$ records how often experts i and j appear together in the same token's top-k route, averaged over probe tokens (self-loops excluded so the metric measures inter-expert collaboration):

$$w_{ij}^{(\ell)} = \frac{1}{N}\sum_{t=1}^{N} \mathbf{1}\{i, j \in \mathrm{top}_k(x_t)\}, \quad i \neq j, \quad (1)$$

where N is the number of probe tokens, drawn from a held-out probing pool of roughly 5,000 tokens per domain (Appendix F.8), and $\mathrm{top}_k(x_t)$ is the router's top-k set for token x_t at layer ℓ. We threshold edges by a retained-edge fraction ρ (default 0.10, with a robustness analysis in Appendix B.2) and run Louvain community detection (Blondel et al., 2008) layer-wise, partitioning experts to maximize modularity. For partition alignment we report **community NMI**, the layer-averaged normalized mutual information (NMI) between the post-training and base partitions on the same probe corpus. We also report modularity Q (Newman, 2006) for absolute clustering strength and ΔQ for its change relative to the base model.

3 Method

3.1 Router Prior Bias (RPB)

RPB is a soft, training-time logit bias derived from a frozen base router and added to the current router's logits during post-training (Figure 2).

Frozen-base prior over Q/A spans. For each training sample x, span $s \in \{Q, A\}$, layer ℓ, and expert e, we precompute the frozen-base routing prior

$$\pi_{x,s,\ell,e} = \mathrm{mean}_{t\in(x,s)}\, p_{\mathrm{base}}(e \mid x_t, \ell), \quad (2)$$

where p_{base} is the frozen base router's softmax gate. A *span* is one of two semantic regions of the training example: **Q** (instruction/prompt) and **A** (response/answer). We use Q/A spans because Q and A induce different base routing distributions (Appendix C), and aggregating them would conflate input-domain with output-format routing.

Training-time bias. Let $c_{x,s,\ell,e} = \mathrm{clip}(\pi_{x,s,\ell,e})$ and let $z_{t,\ell}$ be the router-logit vector across experts at layer ℓ for token t. RPB adds a zero-mean log-prior bias to the logits:

$$b_{x,s,\ell,e} = \log c_{x,s,\ell,e} - \mathrm{mean}_{e'} \log c_{x,s,\ell,e'}, \quad (3)$$

$$z'_{t,\ell} = z_{t,\ell} + \beta\, b_{x,s,\ell}, \quad t \in (x,s). \quad (4)$$

Here $\mathrm{clip}(p)$ clamps probabilities to $[\epsilon/E, C/E]$, where E is the number of experts in the layer (defaults ϵ=0.05, C=5.0), keeping the log-bias finite, and the expert-dimension mean subtraction makes the bias a *relative* expert preference rather than a global logit shift. The bias is broadcast across all tokens in the span, and the prior tensor is frozen so that only the current model parameters train. β controls intervention strength, with β=0 recovering supervised fine-tuning (SFT) with no prior.

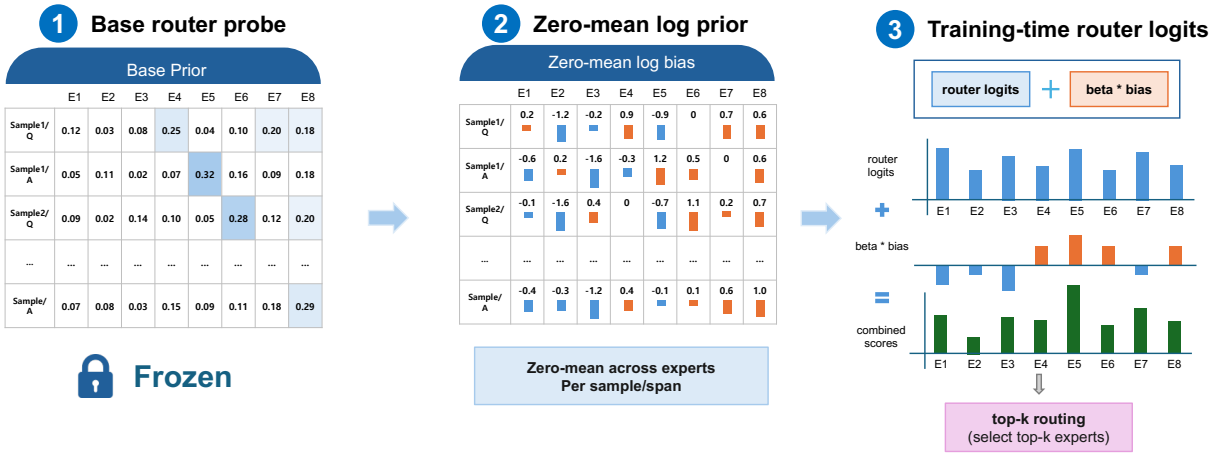


Figure 2: RPB first estimates a frozen-base expert prior for each sample/span and layer, then converts it into a zero-mean log-prior bias across experts. During training, this bias is added to the current router logits with strength β before top-k selection, softly anchoring routing toward the base model. The bias is removed at evaluation, so the measured routing reflects the trained router alone.

Deterministic top-k override (RPB-hard). Since β never removes the router's own contribution to selection, the extreme of the enforcement axis is defined separately, as a control. **RPB-hard** replaces the router logits with the same clipped log-prior as the sole top-k selection score, with gate weights still taken from the unmodified router. Every token in a Q or A span is then forced through the span's aggregate top-k, which removes token-conditioned expert selection. The override applies at training only, and the router is unmodified at evaluation.

3.2 Router-L2

Router-L2 is a router anchoring intervention that uses no sample-conditional prior. It adds to the standard fine-tuning objective a quadratic penalty that keeps each router weight matrix near its frozen base value:

$$L = L_{\mathrm{SFT}} + \lambda \,\mathrm{mean}_{\ell} \left\| W_{\ell} - W_{\ell}^{\mathrm{base}} \right\|_2^2. \quad (5)$$

Only router parameters are anchored: experts, attention, and MLP blocks remain fully trainable, and the router still makes per-token decisions. Unlike RPB, it uses no sample IDs, span labels, or prior tensors, which makes it the cleanest test of whether the sample-conditional prior accounts for the gain. The main experiments sweep $\lambda \in \{10^2, 10^3, 10^4\}$ and report $\lambda{=}10^4$ unless noted, with per-setting results in Appendix A.2.

3.3 Output-Space Router Anchoring

RPB and Router-L2 leave the router's output distribution unconstrained, the space a direct distillation objective would act on, so we add two span-level output-space anchors against the same frozen base-router prior. Let $\pi_{x,s,\ell}$ denote the frozen base prior for sample x, span s, and layer ℓ, and let $p_{\theta,t,\ell}$ be the current router probability for token t in that span. The KL-to-base-router baseline adds

$$\mathcal{L}_{\text{KL-base}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} D_{\mathrm{KL}} \left(\pi_{x(t),s(t),\ell} \,\|\, p_{\theta,t,\ell} \right),$$

where $\mathcal{T}$ is the set of Q/A tokens with a defined span prior. The logit-L2-to-base baseline instead compares centered logits:

$$\mathcal{L}_{\text{logit-L2}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\delta_{t,\ell}\|_2^2,$$
$$\delta_{t,\ell} = \mathrm{center}(z_{t,\ell}) - \mathrm{center}(\log \pi_{x(t),s(t),\ell}).$$

Here $\mathrm{center}(\cdot)$ subtracts the mean across experts, the same centering applied to the RPB bias, and both objectives are training-time only, matching the bias-free protocol used for RPB. Unlike RPB, they pull the router output toward the frozen base target through the loss, so the anchoring never enters the top-k selection step.

4 Experimental Setup

Models. We inspect three base (pretrained) MoE checkpoints: `moonshotai/Moonlight-16B-A3B`,

method	role	constrained object	token-conditioned routing
Baselines			
SFT / NoAux	task loss only	none	preserved
LBL	load-balance baseline	marginal expert load	preserved
Base-router-preserving interventions			
RPB	soft prior intervention	router logits toward base prior	preserved (soft bias)
Router-L2	parameter-space anchor	router weights toward base weights	preserved
KL-to-base	probability-space anchor	router distribution toward base	preserved
Logit-L2	centered-logit-space anchor	centered router logits toward base	preserved
Hard-enforcement and shuffled-prior controls			
RPB-hard	hard base-prior override	base prior forces top-k	removed
within-soft	soft shuffled-prior probe	shuffled community prior, soft	preserved (soft bias)
within-hard	hard within-community override	shuffled prior forces top-k	removed
global-hard	hard global override	global shuffled prior forces top-k	removed

Table 1: Training conditions compared in the main results, grouped by intervention family. *Baselines* fine-tune without (SFT / NoAux) or with marginal-load balancing (LBL); *base-router-preserving interventions* apply different anchoring mechanisms toward the base router; and the *controls* vary the strength of enforcement (RPB-hard) and the content of the prior (the shuffled variants) separately.

`deepseek-ai/DeepSeek-V2-Lite`, and `Qwen/Qwen3-30B-A3B-Base`. DeepSeek-V2-Lite and Moonlight-16B-A3B share the same expert geometry: top-6 selection over 64 routed experts, two always-on *shared experts*, and the same per-expert hidden dimension. They differ in how the router scores those experts, a difference we return to in §5.2.3: Moonlight-16B-A3B uses sigmoid gating with bias-corrected, auxiliary-loss-free top-k selection, whereas DeepSeek-V2-Lite uses softmax scoring with greedy top-k. Qwen3-30B-A3B-Base widens the routed pool to top-8 selection over 128 experts across 48 layers, against top-6 of 64 across 27 layers for the other two, so it varies routing width as well as model family. It supplies base-router statistics as a frozen probe (Appendix C) and is post-trained as a second family in §5.1.3. Hardware and software configuration are in Appendix F.1.

Testbed. Base MoE routers differ in how strongly they separate domains, which bounds the resolution any community-level analysis can achieve. Moonlight-16B-A3B's base routing is far from uniform, with strong Q/A cross-domain divergence, while DeepSeek-V2-Lite's is diffuse and nearly domain-invariant (Appendix D.2), so we center the intervention analysis on the former and treat the latter as a cross-architecture scope condition (Appendix D). Qwen3-30B-A3B-Base enters for the performance claim of §5.1.3, not the community analysis; its base router also separates domains clearly, most sharply on Q spans, though Table 11 measures this by centroid cosine rather than the L_1 statistics used for the other two, so it is not placed on the same sharpness axis.

Datasets. **math20k** and **coding20k** are 20,000-row post-training splits derived from GLM-5.1-Reasoning-1M-Cleaned (Jackrong, 2026): math20k from the `Math` topical subset and coding20k from the `main` subset filtered for fenced code blocks. We use only final completions, so both corpora are (prompt, answer) pairs without an explicit reasoning channel. Pairing the two domains lets us measure in-domain and out-of-domain performance symmetrically, since each domain's out-of-domain set contains the other. Source parsing, the coding-block heuristic, and the deterministic quality scorer are in Appendix F.2.

Evaluation. We evaluate on twelve benchmarks across four task families: math (GSM8K, MATH-500, LiveBench Math), coding (HumanEval, MBPP, LiveBench Coding), general, STEM and professional QA (MMLU, MMLU-STEM, MMLU-Pro, GPQA, GPQA-Diamond), and general reasoning (LiveBench Reasoning). We report $\mathrm{Avg}_{\mathrm{ID}}$ (in-domain mean) and $\mathrm{Avg}_{\mathrm{OOD}}$ (out-of-domain mean), taking the in-domain set to be the post-training domain and the out-of-domain set its complement. Aggregate scores use a nine-benchmark subset of the twelve, written $\mathrm{Overall@9}$, which drops three overlapping MMLU and GPQA variants; Appendix F.4 lists the suite and Appendix G.2 reports every benchmark individually. Routing-state metrics use a held-out slice of the corresponding post-training corpus, disjoint from training tokens. Unless a table notes

Data	Method	ID	OOD
math20k	SFT	29.44 ±0.32	15.65 ±0.38
	LBL	31.91 ±0.10	14.97 ±0.51
	RPB	**45.77** ±0.84	**19.53** ±0.28
coding20k	SFT	28.34 ±1.36	35.83 ±0.49
	LBL	30.73 ±0.13	35.56 ±0.38
	RPB	**41.37** ±0.30	**38.70** ±0.11

Table 2: Moonlight-16B-A3B in-domain and out-of-domain accuracy over the nine-benchmark suite (Appendix F.4; 3 ID / 6 OOD in both blocks), mean ± standard deviation across three training seeds (s42 / s43 / s44) with each seed score an avg@5. RPB improves in-domain adaptation without the out-of-domain cost incurred by re-applied LBL.

otherwise, every condition is trained with three seeds (s42 / s43 / s44) and each checkpoint scored as avg@5 over five evaluation repeats, so tables report seed means with standard deviations (Appendix F.5); sampling, decoding, prompt templates, and per-task answer extraction are in Appendix F.3.

Interventions. The conditions of Table 1 share the protocol above and differ only in their auxiliary objective or router-attached bias (Appendix F.6). **SFT** denotes the task loss alone, with no load-balancing objective, and appears as **NoAux** in Tables 3 and 4.

5 Results

5.1 Re-Applied Load Balancing Underperforms Soft Router Anchoring

None of the evidence here depends on community NMI, whose diagnostic role we take up in §5.2.

5.1.1 Main Results: Moonlight-16B-A3B

On Moonlight-16B-A3B, RPB substantially outperforms both SFT and LBL. On math20k it reaches 45.77 in-domain accuracy against 31.91 for LBL and 29.44 for SFT, and 19.53 out of domain against 14.97 and 15.65 (Table 2). The gain survives disaggregation: RPB attains the best mean on 21 of the 24 (training-dataset, benchmark) cells (Table 16), and all three exceptions fall on coding20k, where SFT scores higher on MMLU and LiveBench Reasoning and LBL is nominally higher on GPQA-Diamond by 0.81 points, well inside the seed noise.

One reading of both effects is that experts outside a token's inherited community are less exposed to the post-training gradient, though we do not measure this directly.

Method	math20k	coding20k
Soft router anchoring		
RPB (span-level logit bias)	**28.27** ±0.16	39.59 ±0.14
Router-L2 (parameter space)	27.69 ±0.28	39.35 ±0.35
KL-to-base (probability space)	27.50 ±0.62	39.27 ±0.33
Logit-L2 (centered logit space)	27.24 ±0.77	**39.81** ±0.67
Baselines		
NoAux / SFT (no load balancing)	20.25 ±0.29	33.34 ±0.77
LBL (re-applied)	20.61 ±0.33	33.95 ±0.22

Table 3: Four anchoring objectives acting in different spaces perform comparably on Moonlight-16B-A3B, with Router-L2 at $\lambda{=}10^4$. Values are Overall@9, the mean over the nine-benchmark suite of Appendix F.4, reported as mean ± standard deviation across three training seeds (s42 / s43 / s44) with each seed score an avg@5. Best per column in bold. The winner changes between the two training corpora, so no single anchoring form is superior on both.

5.1.2 High Performance Does Not Depend on the Anchoring Form

If the gain depended on the particular form of RPB's bias, anchoring the router in a different space should not reproduce it. Router-L2 (§3.2) is the first such test, replacing the sample-conditional logit bias with a loss-integrated penalty on the router weights themselves; on Moonlight-16B-A3B it matches RPB to within 0.6 points on both corpora, so the improvement is not specific to the bias form. Both objectives nonetheless leave the router's output distribution unconstrained, so we add the two output-space anchors of §3.3, which between them cover the natural router-distillation targets, and report all four under an identical protocol (Table 3). They fall within about one point of one another on math20k and half a point on coding20k, against a gap of roughly seven and six points to either baseline, and their ordering does not survive a change of corpus: on coding20k the centered logit-L2 anchor outperforms RPB (39.81 ± 0.67 against 39.59 ± 0.14), while on math20k the ordering reverses. Reaching this level therefore does not require RPB's sample-conditional prior, only that the router be held near its base routing while remaining trainable.

5.1.3 Generality Across Model Family and Corpus

We vary the model family and the corpus in turn, under the same protocol (Table 4). The first perturbation is a change of MoE family: we post-train Qwen3-30B-A3B-Base on both corpora (upper block). Re-applied LBL is again the weakest

	Method	math20k	coding20k
Qwen3-30B	RPB	**57.58** ±0.74	63.13 ±0.08
	Router-L2	57.39 ±0.40	63.93 ±0.21
	NoAux	56.96 ±0.84	**64.10** ±0.51
	LBL	54.94 ±1.87	62.97 ±0.21
Moonlight-16B-A3B, OpenR1-Math-220k			
	RPB	**41.40** ±0.28	
	NoAux	40.44 ±0.34	
	LBL	31.20 ±0.44	

Table 4: Generality of the advantage over re-applied LBL. The upper block is Qwen3-30B-A3B-Base on math20k and coding20k, and the lower block is Moonlight-16B-A3B on OpenR1-Math-220k. Values are Overall@9, mean ± standard deviation across three training seeds. Re-applied LBL is the weakest condition in every block, while the separation between the anchors and the no-load-balancing baseline is model- and corpus-dependent.

	math20k		coding20k	
Condition	**Overall@9**	**NMI**	**Overall@9**	**NMI**
RPB (unshuffled reference)	**28.27**	0.631	**39.59**	0.629
within-soft	27.23	0.632	38.96	0.632
within-hard	18.77	0.634	24.87	0.636
global-hard	3.59	**0.642**	17.96	**0.650**

Table 5: Shuffled-prior controls on Moonlight-16B-A3B. Holding community labels fixed and switching enforcement from soft to hard (within-soft → within-hard) costs more than scrambling the labels while keeping enforcement soft (RPB → within-soft). Community NMI moves in the opposite direction to performance, since the global-hard control attains the highest retention on both corpora while performance falls sharply. Overall@9 columns are three-seed means (s42 / s43 / s44); NMI columns are three probed seeds for RPB and two (s42 / s43) for the shuffled controls, following Appendix F.5. Per-benchmark breakdown in Appendix G.3.

condition on both, so the ordering reproduces on a second family, but only the math20k margin is resolvable at our seed budget: LBL trails RPB by 2.64 points there, while the 0.16-point gap on coding20k sits inside the noise. The advantage over the no-load-balancing baseline does not carry over at all: RPB, Router-L2 and NoAux fall within a band narrower than the seed noise on math20k, and on coding20k NoAux outperforms every other condition, both anchors included.

The second is a change of corpus, since agreement between two splits of one source says little about corpus dependence. On OpenR1-Math-220k, an independently constructed mathematics corpus yielding roughly 94k examples after the same preprocessing (lower block), re-applied LBL costs 10.20 points of Overall@9 against RPB, more than twenty times the seed noise, while the gap to the no-load-balancing baseline is 0.96, small but outside the combined standard deviation.

The two perturbations therefore split the claim: the disadvantage of re-applied LBL survives both; the advantage over unconstrained fine-tuning survives neither.

5.2 Routing Retention as a Diagnostic of Soft Router Anchoring

We now ask which level of routing retention tracks the performance ordering.

5.2.1 Community Structure as a Footprint of Soft Enforcement

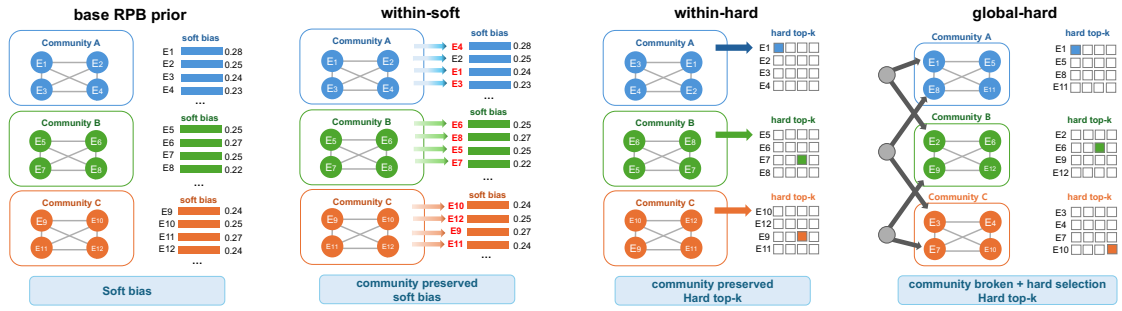
The three levels of §2 do not separate the methods equally (Table 15). SFT, LBL and RPB all fall in the same TVD and JSD range, so per-expert utilization carries no information about which intervention was applied. Top-k overlap does separate LBL from the rest, but it places SFT between the two anchors on coding20k (0.482, against 0.481 for RPB and 0.486 for Router-L2), so only community-level retention isolates the soft-anchoring family on both corpora.

Separating the methods does not, however, make community NMI the quantity to maximize. It saturates early under a β sweep, and replacing the soft bias with the hard assignment of §3.1 pushes it above the soft maximum while performance falls well below it, on coding20k even below unanchored fine-tuning (Appendix A.1), so it is a footprint left by soft enforcement at training time rather than an optimization target.

5.2.2 Interventional Controls: Soft versus Hard Enforcement

A positive RPB result does not on its own separate the content of the prior from the strength with which it is enforced. We vary the two axes independently (Figure 3) with three controls: **within-soft** (within-community shuffle, soft logit bias), **within-hard** (within-community shuffle, hard top-k override), and **global-hard** (across-community shuffle, hard override). The within-community shuffle is a substantial perturbation rather than a nominal one, changing 46–60% of the prior's top-k entries by an amount comparable in magnitude to the router logits themselves (Appendix E).

Three observations follow from Table 5. First, *within-soft stays within about one point of RPB*, with community NMI matched as well (Table 5),



	base	within-soft	within-hard	global-hard
Community boundary	preserved	preserved	preserved	broken
Expert identity	intact	shuffled within community	shuffled within community	shuffled globally
Routing behavior	soft bias	soft bias	hard top-k	hard top-k

Figure 3: Three shuffle controls applied to the base RPB prior over experts E_i. In **within-soft**, expert identities are permuted only within each detected community, while the shuffled prior is still applied as a soft router bias, so community structure is preserved and token-conditioned routing remains soft. In **within-hard**, the same within-community permutation is converted into hard top-k expert selection, preserving community membership but removing soft router choice. In **global-hard**, expert identities are permuted across communities and then used for hard top-k selection, so the community structure is broken in addition to soft routing being replaced by hard assignment.

so the prior need not name individual base experts and community-level anchoring suffices. Second, *within-hard reduces performance sharply despite the same community-preserving prior content*, leaving enforcement strength as the only axis that changed and hence as the operative one. Third, *global-hard reduces performance further still*, once community membership is corrupted as well; its community NMI is nonetheless the highest of the four conditions, which reflects the limited resolution of NMI on weakly clustered graphs rather than genuine retention, and is why we validate the community object against a marginal-preserving null model and report modularity Q alongside it (Appendix B.1).

5.2.3 Scope Condition: DeepSeek-V2-Lite

The diagnostic presupposes a base router with community structure to retain, and DeepSeek-V2-Lite is a model where that fails: its routing is close to uniform and nearly domain-invariant (§4); we report this as a measured property of the base checkpoint rather than attribute it to any single design choice.

Community NMI then has little structure to compare across conditions and fails to separate the interventions as it does on Moonlight-16B-A3B (Appendix D). Performance still improves under RPB there, and the soft-against-hard dissociation still reproduces, so we read the flat NMI as the diagnostic behaving as specified rather than as a failure of the intervention, and state it only for base models whose routing is non-uniform enough to induce community structure.

6 Discussion

The four anchoring objectives act on router parameters, on router logits, or on the routing distribution, yet perform comparably, and what they share is that each keeps the router close to its base routing while leaving it trainable. LBL, constraining only the marginal load, does not reach that level, while the hard variants push the constraint past it, reaching comparable or higher community NMI while performance declines. That the space of the anchor makes no consistent difference while its softness makes a large one places the effect in the strength of the constraint rather than in the quantity constrained. The ST-MoE z-loss reaches the same conclusion from the opposite direction, regularizing the router with no reference to the base model and trailing even unanchored fine-tuning on both corpora (Appendix A.3).

The dissociation between community NMI and performance carries a warning beyond MoE routing. Inside the soft family, retained community

structure covaries with accuracy closely enough to suggest an objective worth maximizing, yet hard enforcement attains retention at or above the soft maximum while performance drops sharply. A proxy of the form "distance from the base model" is therefore informative only over the range in which it was calibrated, and optimizing it directly is what pushes it outside that range.

At fixed hard enforcement, shuffling expert identities across communities rather than within one costs a further 15.2 points on math20k and 6.9 on coding20k, so post-training reuses specialization at the granularity at which experts collaborate.

Two design constraints follow. The anchoring target should be the inherited co-activation structure rather than the per-expert load distribution, since flattening the latter removes the property that supports out-of-domain retention; and the anchoring should stay soft, since the interventional controls locate the failure at enforcement strength rather than at prior content. Methods that freeze or hard-select experts during adaptation (Wang et al., 2024b; Li et al., 2025; Eo et al., 2025) fix their enforcement strength by construction, so graded variants are worth testing against the range Router-L2 spans. What determines whether the diagnostic applies is how sharply the base router already separates domains rather than the expert architecture: shared-expert designs (Dai et al., 2024; Guo et al., 2025) appear on both sides of our comparison, so that design choice does not by itself settle the question.

7 Related Work

MoE fine-tuning and routing control. Sparse MoE pretraining stabilizes the expert pool through utilization-oriented routing objectives (Lepikhin et al., 2020; Fedus et al., 2021; Zhou et al., 2022), most commonly an auxiliary load-balancing loss (LBL), with refinements such as global-batch variants (Qiu et al., 2025) and auxiliary-loss-free or dynamic-bias balancing (Wang et al., 2024a; DeepSeek-AI, 2024b); all of these target pretraining dynamics, where no inherited routing exists yet to preserve. The closest work to ours instead asks which experts to update, freeze, or select during adaptation: ESFT (Wang et al., 2024b) freezes experts by relevance, DES-MoE (Li et al., 2025) targets dynamic specialization, and MoCE (Eo et al., 2025) routes within a selected group, close in spirit to soft anchoring. Practitioner toolchains have converged on similar empirical choices, dropping load balancing or router training altogether during post-training (NVIDIA, 2025; Unsloth AI, 2025). We complement both lines by targeting the router itself, constraining how routing changes during post-training rather than reshaping the expert pool.

Expert specialization. A parallel line treats experts as *specialized*. Architectural designs separate shared and fine-grained routed experts (Guo et al., 2025; Dai et al., 2024); Moonlight-16B-A3B and DeepSeek-V2-Lite both adopt that design, so the contrast in §5.2.3 turns on their router scoring rules rather than on this axis. Probe studies find semantic and multilingual signal in routing (Lo et al., 2024; Bai et al., 2025), and frozen-checkpoint analyses target compression (Lu et al., 2024; Hu et al., 2026). A complementary view treats expert *collaboration*, meaning which experts co-activate on the same token, as the analysis target, through co-activation matrices and graph-structured MoE (Tang et al., 2026; Nguyen-Nhat et al., 2025). We adopt this collaboration view but treat the co-activation graph as a *state variable* whose retention is measurable after post-training, using standard network-science tools: modularity Q (Newman, 2006) with Louvain (Blondel et al., 2008) and Leiden (Traag et al., 2018) community detection.

8 Conclusion

Re-applying the pretraining load-balancing loss degrades downstream performance relative to leaving the inherited routing softly anchored, and the effect survives a change of model family and of training corpus. Four anchoring objectives acting in different spaces perform comparably on Moonlight-16B-A3B, so **RPB is one competitive implementation of soft router anchoring rather than a uniquely necessary one**. Subject to two boundaries, that the advantage over a no-load-balancing baseline is model-dependent and that community NMI is informative only where the base router carries community structure to begin with, these findings move the target of MoE post-training stabilization from re-imposing uniform expert utilization to preserving the routing the base model has already learned.

Limitations

The community-mediated evidence is Moonlight-centered. We state the diagnostic with a scope condition rather than as a general property because of DeepSeek-V2-Lite's diffuse routing distribution,

and on Qwen3-30B-A3B-Base the diagnostic and the performance claim come apart, since the ordering against re-applied LBL is preserved while the community-level reading is not the operative evidence there. The within-community shuffle result should likewise be read as a statement about the communities Louvain detects in Moonlight-16B-A3B, not as evidence that they are dense, semantically coherent modules, since we do not characterize their internal composition. Establishing how widely the community reading applies would require base models sampled across a spectrum of routing sharpness, and further MoE families to test whether the same footprint appears under different community structures; neither is within our scope.

The comparison against a no-load-balancing baseline is the weaker half of our performance result and should not be read as stronger than it is. Soft anchoring outperforms it by a wide margin on Moonlight-16B-A3B and by a small but reliable one on OpenR1-Math-220k, yet on Qwen3-30B-A3B-Base the two are indistinguishable on math20k and the no-load-balancing baseline is superior on coding20k. We report the reversal rather than restricting the comparison, but we cannot presently say which property of a base model predicts the direction of the effect.

The fine-tuning data are intentionally controlled 20k-example splits from a single source corpus, GLM-5.1-Reasoning-1M-Cleaned. This design makes the in-domain and out-of-domain contrast between math and code symmetric and keeps the routing interventions comparable, but it does not establish that the same hyperparameter range or effect size will hold under larger or more heterogeneous instruction mixtures.

Router-L2 shows that the improvement is not specific to an additive logit bias, and the output-space anchors cover the direct-distillation alternative. None of the four objectives separates anchoring strength from anchoring target within a single objective. Each fixes a target and varies strength, so we cannot rule out that a different target would move the point at which adaptation and retention are best traded off. We also do not have a criterion that predicts which anchoring form will lead on a given corpus, only the observation that the ordering changes across corpora.

We focus on non-reasoning base models, with math20k and coding20k as correspondingly non-reasoning QA splits. Reasoning-trace post-training with long chains of thought is not covered, and the structural retention dynamics may differ when the gradient signal is heavily weighted toward long-form generation. In such settings the relevant object may not be a static co-activation graph but *sequential routing transition communities*, defined over the time-ordered pattern of expert-set transitions across reasoning steps. Constructing those transition graphs and re-running the shuffled-control protocol against them is a natural next step.

References

Jun Bai, Minghao Tong, Yang Liu, Zixia Jia, and Zilong Zheng. 2025. Understanding and leveraging the expert specialization of context faithfulness in mixture-of-experts LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21927–21942, Suzhou, China. Association for Computational Linguistics.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 37:3896–3915.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, Bangkok, Thailand. Association for Computational Linguistics.

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

DeepSeek-AI. 2024b. Deepseek-v3 technical report. *ArXiv*, abs/2412.19437.

Sugyeong Eo, Jung Jun Lee, Chanjun Park, and Heuiseok Lim. 2025. Mixture-of-clustered-experts: Advancing expert specialization and generalization in instruction tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14201–14212, Suzhou, China. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv*, abs/2101.03961.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Xinye Cao, Sicong Leng, Qimei Cui, and Xudong Jiang. 2025. Advancing expert specialization for better moe. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wentao Hu, Mingkuan Zhao, Shuangyong Song, Xiaoyan Zhu, Xin Lai, and Jiayin Wang. 2026. Mosaic pruning: A hierarchical framework for generalizable pruning of mixture-of-experts models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 21885–21893.

Jackrong. 2026. Glm-5.1-reasoning-1m-cleaned.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Z. Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *ArXiv*, abs/2006.16668.

Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. 2025. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain moe adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. Muon is scalable for llm training. *Preprint*, arXiv:2502.16982.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2024. A closer look into mixture-of-experts in large language models. *ArXiv*, abs/2406.18219.

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. In *Annual Meeting of the Association for Computational Linguistics*.

Mark E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103 23:8577–82.

Minh-Khoi Nguyen-Nhat, Rachel S. Y. Teo, Laziz U. Abdullaev, Maurice Mok, Viet-Hoang Tran, and Tan M. Nguyen. 2025. Modeling expert interactions in sparse mixture of experts via graph structures. *ArXiv*, abs/2510.16411.

NVIDIA. 2025. Adapting Megatron Bridge in reinforcement learning frameworks. https://docs.nvidia.com/nemo/megatron-bridge/latest/bridge-rl-integration.html#best-practices-and-pitfalls. Accessed 2026-05-24.

Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5005–5018.

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

Yuanbo Tang, Naifan Zhang, Yan Tang, Meixuan Chen, Shuhan Huang, Tingyu Cao, and Yang Li. 2026. Specialization through collaboration: Understanding expert interaction in mixture-of-expert large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2339, Rabat, Morocco. Association for Computational Linguistics.

Vincent Antonio Traag, Ludo Waltman, and Nees Jan van Eck. 2018. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9.

Unsloth AI. 2025. Fine-tune qwen3.5 (moe fine-tuning 35b / 122b). https://unsloth.ai/docs/models/qwen3.5/fine-tune#moe-fine-tuning-35b-122b. Accessed 2026-05-24.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. 2024a. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *ArXiv*, abs/2408.15664.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. 2024b. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 784–801.

Yanqi Zhou, Tao Lei, Han-Chu Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. *ArXiv*, abs/2202.09368.

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

Appendices

A Further Controls: β Range and z-loss

A.1 β Operating Range and the Hard-Assignment Ceiling

Table 6 sweeps β, which controls the strength of the RPB prior. As β rises, community NMI rises and saturates by β=0.5, while performance follows a unimodal shape peaking near β=0.75 to 1.0. The hard end of the sweep (**RPB-hard**) uses the same prior as the sole top-k selection score, with contribution weights still computed from router logits. It raises community NMI slightly above the soft maximum yet reduces performance on both corpora, falling below even β=0 on coding20k, which is the dissociation §5.2.1 builds on.

A.2 Router-L2 λ Sensitivity

Table 7 completes the sweep announced in §3.2. Across two orders of magnitude the anchoring strength moves the operating point only slightly: the three settings span 0.25 points on math20k, inside the seed spread of the λ=10^4 row, and 1.0 point on coding20k. The ordering also reverses between corpora, with the weakest anchor best on coding20k and the strongest best on math20k, which is the same corpus-dependence the four anchoring objectives show in §5.1.2. We report λ=10^4 in the main tables because it is the only setting trained with three seeds, not because it is the sweep optimum.

A.3 ST-MoE z-loss

Table 8 reports the ST-MoE z-loss (Zoph et al., 2022), included as a baseline regularizer drawn from the MoE literature rather than as an anchoring objective. It penalizes large router logits to stabilize training and carries no reference to the base router. Its rows come from a separate set of runs, so we read the direction of the gaps below rather than their size.

In every case the z-loss falls below RPB, and on Moonlight-16B-A3B it also falls below unanchored fine-tuning on both corpora, whether the SFT figure is taken over the nine-benchmark suite or over all twelve. The negative result supports the broader point, since it shows that regularizing the router is not sufficient on its own: the regularizer has to act toward the inherited routing structure rather than toward a generic property of the routing distribution.

B Validity and Robustness of the Community Structure

B.1 Marginal-Preserving Null Model

Table 9 verifies that the detected communities are not merely artifacts of marginal expert usage. We compare each observed co-activation graph against a marginal-preserving null model, namely 100 multinomial-resampled graphs per `(run, cluster, layer)` whose expected edge weights are proportional to the product of expert marginals, which preserves each expert's marginal mass while removing pairwise co-selection. Real graphs beat this null on modularity in 100% of cells for every condition, with Q margins of 0.132 to 0.210. The NMI margins are smaller throughout, at 0.037 to 0.091, which reinforces that community NMI is diagnostic rather than explanatory. We use the null model only as a validity check for the community object, not as a method-ranking metric.

B.2 Threshold and Detector Sweeps

Table 10 varies the two free choices in the community construction of §2. The edge fraction ρ moves the absolute level of community NMI substantially, since retaining fewer edges leaves a sparser graph whose partition is easier to reproduce, but it leaves the method ordering RPB $\approx$ Router-L2 $>$ SFT $>$ LBL unchanged at every setting on both corpora. Substituting Leiden for Louvain at otherwise-matched parameters changes no value by more than 0.009, and dispersion across detector seeds $\{0, 1, 2, 42\}$ averages 0.031 community NMI and never exceeds 0.050, so it stays below the RPB-to-LBL gap at every ρ in the sweep. The separate sweep behind the within-community shuffle, over $\rho \in \{0.05, 0.10, 0.15, 0.20\}$ and permutation seeds $\{0, 1, 2\}$, keeps within-community preservation at 0.986 community NMI on average, never below 0.957 and with standard deviation 0.011 across all combinations, so the perturbation of §5.2.2 preserves community membership by construction rather than by chance.

C Base-Router Heterogeneity Probes

To check whether the Q/A split used by RPB is specific to Moonlight-16B-A3B and DeepSeek-V2-Lite or also appears in a newer MoE family, we additionally probed Qwen3-30B-A3B-Base. In this appendix the model serves only as a frozen

β / variant	math20k		coding20k	
	Community NMI	Overall@9	Community NMI	Overall@9
0 (SFT)	0.575	20.25	0.588	32.68
0.5 constant	0.624	27.41	0.630	38.37
0.75 constant	0.629	28.11	0.629	**40.04**
1.0 constant	0.631	**28.44**	0.629	39.66
RPB-hard	0.634	24.08	0.642	31.54

Table 6: Operating range of RPB across β on Moonlight-16B-A3B, with the hard end of the sweep (RPB-hard) for contrast. Community NMI is the layer-mean community-level normalized mutual information against the base partition, and Overall@9 is the nine-benchmark aggregate of Appendix F.4. Single training seed (s42), each score an avg@5 over five evaluation repeats; all comparisons drawn from this table are therefore seed-matched. The β=0 row is the single-seed counterpart of the NoAux baseline of Table 3, whose three-seed mean differs slightly, so the two should not be read against each other.

λ	math20k	coding20k
10^2	27.44†	**40.35**†
10^3	27.51†	39.64†
10^4	**27.69** $\pm$0.28	39.35 $\pm$0.35

Table 7: Router-L2 anchoring strength on Moonlight-16B-A3B, reported as Overall@9. The λ=10^4 rows are three-seed means $\pm$ sample standard deviation (s42 / s43 / s44); †the two weaker settings are single-seed (s42), so the bolded coding20k entry should not be read as a significant win. Each seed score is itself an avg@5 over five evaluation repeats.

Setting	ST-MoE Avg
Moonlight math20k	16.33
DeepSeek math20k	9.59
Moonlight coding20k	30.05
DeepSeek coding20k	17.57

Table 8: ST-MoE z-loss (Zoph et al., 2022) results on Moonlight-16B-A3B and DeepSeek-V2-Lite, included as a baseline regularizer drawn from the MoE literature rather than as an anchoring objective. Values are the twelve-benchmark mean at a single training seed (s42), each score an avg@5 over five evaluation repeats, so they are on the same scale as the **Avg** column of Table 16 rather than the Overall@9 of the main tables.

base router, downloaded directly from HuggingFace, and its post-training results are reported separately in §5.1.3. It is a non-reasoning probe in our setup, since no probe row carries a separate reasoning trace, so every sequence decomposes into the Q and A spans of §3.1 alone. Table 11 reports the two spans for Moonlight-16B-A3B and Qwen3-30B-A3B-Base. Q tokens separate the math, code, and general probe clusters more sharply than A tokens do in both models, on all three pairwise cosine similarities, and code is the most separated cluster on Moonlight-16B-A3B. The pattern holds

Dataset	Method	ΔQ	ΔNMI
math20k	SFT	0.140	0.053
	LBL	0.210	0.048
	RPB	0.132	0.050
	RPB-hard	0.140	0.039
coding20k	SFT	0.149	0.052
	LBL	0.174	0.091
	RPB	0.132	0.046
	RPB-hard	0.161	0.037

Table 9: Marginal-preserving null-model comparison on Moonlight-16B-A3B. ΔQ and ΔNMI are the margins of the observed co-activation graph over 100 multinomial-resampled null graphs per (`run`, `cluster`, `layer`), at edge fraction $\rho = 0.10$ and a single probed seed (s42). Every condition beats the null on modularity in 100% of cells, and the modularity margin is consistently larger than the NMI margin, which is why we read community NMI as a diagnostic rather than a sufficient explanation. The margins do not order the methods the way performance does, so this check validates the community object rather than ranking the interventions.

on Qwen3-30B-A3B-Base despite its larger routed pool of 128 experts, although its A spans retain more domain separation and draw on a broader expert fraction than Moonlight-16B-A3B's, so the split RPB conditions on is not an artifact of a single model family.

D DeepSeek-V2-Lite as a Scope Condition for the Diagnostic

D.1 Cross-Architecture Robustness Check

DeepSeek-V2-Lite has a considerably more diffuse routing distribution than Moonlight-16B-A3B, which makes it a useful cross-architecture robustness test: the performance claim and the community-level diagnostic can be checked separately on it. RPB still improves performance, most

Dataset	Method	Community NMI at ρ 0.05	0.10	0.20
math20k	SFT	0.729	0.580	0.410
	LBL	0.709	0.547	0.393
	RPB	**0.752**	**0.634**	**0.493**
	Router-L2	0.750	0.628	0.488
coding20k	SFT	0.734	0.590	0.444
	LBL	0.689	0.541	0.408
	RPB	0.748	0.629	**0.497**
	Router-L2	**0.752**	**0.632**	0.493

Table 10: Sensitivity of community NMI to the edge-fraction threshold on Moonlight-16B-A3B, at a single probed seed (s42) with Louvain at resolution $\gamma = 1.0$. The absolute level falls as ρ grows and the graph densifies, but the ordering of the four methods is preserved at every threshold. Leiden at matched parameters agrees with Louvain to within 0.009 on every cell, and dispersion over detector seeds $\{0, 1, 2, 42\}$ averages 0.031 and never exceeds 0.050. The main text uses $\rho = 0.10$. Because these are single-seed values they differ slightly from the three-seed means of Table 15. Bold marks the best mean per column within each dataset block.

clearly out of domain on math20k, but community NMI does not separate the methods at all, with all four fine-tuning conditions clustering near NMI ≈ 0.49 (Table 12). Appendix D.2 analyzes the underlying mechanism, namely a diffuse and nearly domain-invariant base routing distribution.

The dependence on soft, token-conditioned routing remains visible even without community-level separation. Under hard span-level control (RPB-hard), performance drops sharply, from 16.12 to 9.44 overall on math20k and from 22.58 to 16.92 overall on coding20k, with in-domain accuracy falling from 22.88 to 9.49.

Two readings follow. The community-mediated channel is model-dependent, since Moonlight-16B-A3B supports the full community dissociation while DeepSeek-V2-Lite shows that performance gains can occur without strong community NMI separation when the base routing distribution is diffuse. The dependence on soft rather than hard enforcement, by contrast, holds on both post-trained models, though two models establish a consistent pattern rather than a general property.

D.2 Why Community NMI Lacks Separation on DeepSeek-V2-Lite

The absence of NMI separation on DeepSeek-V2-Lite (Appendix D, Table 12) reflects three properties of its base routing distribution. *(i) Diffuse routing distribution.* The base routing sharpness $\|p - U\|_1$, the L_1 distance between the layer-averaged expert distribution p and the uniform distribution U over routed experts, is 0.1399 on math20k for DeepSeek-V2-Lite against 0.5445 for Moonlight-16B-A3B, so its base distribution sits much closer to uniform. *(ii) Low cross-domain divergence.* $\|p_{\text{math}} - p_{\text{code}}\|_1 = 0.0724$ for DeepSeek-V2-Lite against 0.3735 for Moonlight-16B-A3B, so its routing preferences are nearly domain-invariant. *(iii) Router scoring rule.* The two base routers share their expert geometry, both using top-6 of 64 routed experts alongside two shared experts at the same per-expert hidden dimension, and differ instead in how they score those experts: Moonlight-16B-A3B uses sigmoid gating with bias-corrected, aux-loss-free top-k selection, while DeepSeek-V2-Lite uses softmax scoring with greedy top-k. We did not isolate this difference experimentally, so we report it as a candidate explanation for (i) and (ii) rather than a demonstrated cause.

The graph-level consequence is that the DeepSeek-V2-Lite co-activation graphs are naturally more uniform and less clustered, and Louvain on a near-uniform graph cannot return a partition sharply different from the base one whatever the fine-tuning method. All four fine-tuned DeepSeek-V2-Lite variants land within 0.02 of one another at NMI ≈ 0.49, while the Moonlight-16B-A3B variants span a much wider band, so community NMI cannot separate methods when the base community structure is itself weak.

E Prior-Side Validation of Shuffled-Prior Controls

2×2 design. The three controls used in §5.2.2 sit on a 2×2 grid of *shuffle scope* against *enforcement strength*. **Within-community shuffle** permutes expert identities inside each detected base community, so a token that would have been routed to community C is still routed inside C and only its identity within C is scrambled. **Global shuffle** permutes expert identities across all communities, so both community membership and within-community identity are corrupted at once. Each scope is paired with a **soft** application, where the shuffled prior enters as a soft logit bias and token-conditioned routing stays intact, or a **hard** application, where the shuffled prior forces the Q/A-span aggregate top-k and removes token-conditioned routing. The three reported cells, *within-soft*, *within-hard*, and *global-hard*, together isolate the soft against hard

Model	Span	math–code cos.	math–general cos.	code–general cos.	eff. expert frac.
Moonlight-16B-A3B	Q	0.438	0.790	0.362	0.202–0.260
Moonlight-16B-A3B	A	0.788	0.943	0.745	0.245–0.281
Qwen3-30B-A3B-Base	Q	0.409	0.697	0.378	0.228–0.257
Qwen3-30B-A3B-Base	A	0.561	0.800	0.525	0.348–0.404

Table 11: Q/A span routing structure in Moonlight-16B-A3B and Qwen3-30B-A3B-Base, measured from mean top-k activation frequency (`P_topk`) on the math/code/general probe clusters. Cosine similarity is computed between domain centroids after averaging over samples and layers, so lower values indicate stronger domain separation. Effective expert fraction is $\exp(H)/E$ averaged over the listed domains, where H is the entropy of the mean top-k activation frequency over the E routed experts, so larger values indicate broader expert usage. All quantities are measured from the frozen base checkpoints, so no training seeds are involved.

Dataset	Method	ID	OOD	Avg	Comm. NMI
math20k	Base	3.77	17.52	14.08	1.000
	SFT	13.55	14.42	14.21	0.490
	LBL	13.43	15.69	15.12	0.487
	RPB	14.11	16.79	16.12	0.498
	RPB-hard	11.47	8.76	9.44	0.503
coding20k	Base	17.01	13.11	14.08	1.000
	SFT	21.92	21.31	21.46	0.488
	LBL	21.86	20.99	21.21	0.493
	RPB	22.88	22.49	22.58	0.497
	RPB-hard	9.49	19.40	16.92	0.495

Table 12: DeepSeek-V2-Lite cross-architecture robustness check. Single training seed (s42), each score an avg@5 over five evaluation repeats. Unlike the Moonlight-16B-A3B tables, the **ID**, **OOD**, and **Avg** columns here aggregate all twelve benchmarks rather than the nine-benchmark suite, so they cannot be recomputed from Table 17, which lists the nine. **Comm. NMI** is community NMI against the base partition.

switch at fixed community preservation (within-soft against within-hard) and the within against global scope at fixed hardness (within-hard against global-hard).

Outcome statistic vs. intervention strength. Two measurements must not be conflated here. The bias-free probe of the eventual checkpoint shows within-soft and unshuffled RPB landing at near-identical top-k overlap to base, which is an *outcome statistic* about the trained model's routing behavior. The prior-side measurements below instead report *intervention strength*, the perturbation applied to the prior tensor before training ever begins.

E.1 Prior-Side Intervention Strength

The validation script compares the *original* unshuffled RPB prior against the *shuffled* prior at a fixed clamp and β configuration. The prior tensor is `prior_qa`, of shape $N_{\text{samples}} \times 2 \times L \times E$ over the Q and A spans (Appendix F.7). For each (sample, span, layer) we measure three quantities: top-k overlap between the two priors' top-k sets, with k matched to the model's routing top-k; the per-element $|\Delta b|$ at the 90th percentile of the *effective* log-prior bias actually added to the router logits; and the fraction of clamp upper-bound activations.

The within-community shuffle changes $\sim$46% of the prior's top-k entries on math20k and $\sim$60% on coding20k, with a 90th-percentile $|\Delta b|$ in the effective log-prior bias of $\sim$0.47 and $\sim$0.82 respectively. Set against the empirical per-row bias standard deviation of $\sim$0.62 and $\sim$0.65 (Appendix F.7), the perturbation is on the order of one within-row standard deviation, that is, a change at the scale of the logits themselves rather than a nominal one. Global shuffle changes $\sim$90% of top-k entries with a 90th-percentile $|\Delta b|$ of 1.5–1.6, saturating the intervention as a strong negative control should. Clamp upper-bound activation is low in both domains (1–2%), which rules out the reading that the shuffle is silently suppressed by the `clamp(p, eps/E, c/E)` step. The deflationary objection, that the within-community shuffle does not actually change the prior, is therefore not supported on the prior side.

E.2 Hardness vs. Shuffle Scope Decomposition

The within-hard and global-hard rows of Table 14 separate two effects that are otherwise easy to confuse, the hardness of the constraint and the scope of label corruption. The table reports the unweighted mean of the in-domain and out-of-domain figures, $(\text{ID}+\text{OOD})/2$, together with the Δ against the soft RPB reference computed from the same column, so every entry is recomputable from the columns shown.

Reference rows mix seed budgets, since the soft RPB reference and the shuffled controls are three-seed means (§5.1.1, §5.2.2) while RPB-hard is single-seed (Appendix A.1). Cross-protocol noise

Dataset	Variant	prior top-k overlap	top-k change rate	effective log-bias $\lvert\Delta b\rvert$ p90	clamp upper-bound frac.	Verdict
math20k	within-community	0.539	0.460	0.471	0.021	meaningful intervention
math20k	global	0.104	0.896	1.455	0.021	strong negative control
coding20k	within-community	0.397	0.604	0.822	0.011	meaningful intervention
coding20k	global	0.094	0.906	1.581	0.011	strong negative control

Table 13: Prior-side intervention strength of the within-community and global shuffles, measured on the RPB prior tensor before any training, so the values do not depend on training seeds. The overlap and change-rate columns are complementary by construction. The remaining columns are the 90th percentile of the per-element change in the effective log-prior bias and the fraction of entries held at the clamp upper bound.

Dataset	Method	Prior	Constraint	ID	OOD	(ID+OOD)/2	Δ vs. RPB
math20k	RPB (unshuffled reference)	base	soft	45.77	19.53	32.65	reference
math20k	RPB-hard (unshuffled)†	base	hard	39.94	12.18	26.06	hard constraint alone: -6.59
math20k	within-hard	within-shuffled	hard	30.70	12.81	21.75	hard constraint + within-scramble: -10.90
math20k	global-hard	global-shuffled	hard	3.97	3.40	3.68	hard constraint + global-scramble: -28.97
coding20k	RPB (unshuffled reference)	base	soft	41.37	38.70	40.04	reference
coding20k	RPB-hard (unshuffled)†	base	hard	33.90	28.04	30.97	hard constraint alone: -9.07
coding20k	within-hard	within-shuffled	hard	23.14	25.73	24.44	hard constraint + within-scramble: -15.60
coding20k	global-hard	global-shuffled	hard	12.91	20.48	16.69	hard constraint + global-scramble: -23.35

Table 14: Decomposition of prior hardness and shuffle scope on Moonlight-16B-A3B. The RPB reference and the shuffled controls are three-seed means over the same nine-benchmark set as Table 2. †RPB-hard is single-seed (s42) from the β sweep of Appendix A.1 and is reported for contrast only.

is on the order of a few tenths of a point and the gaps in the Δ column are at least an order of magnitude larger, so the qualitative ordering is unaffected.

F Reproducibility Protocol

F.1 Compute Environment

All post-training runs execute on 16 nodes of 8 NVIDIA H200 GPUs, for a world size of 128, under Megatron-Bridge. The parallel degrees are tensor 2, pipeline 1, context 1, expert 8, and expert-tensor 1, giving a data-parallel width of 64; sequence parallelism is enabled, and selective activation recomputation covers the layernorm, MoE, MoE-activation, and core-attention modules. Training runs in mixed `bf16` on `torch` 2.10 with CUDA 12.9 and NCCL 2.27. Evaluation runs separately under vLLM 0.10.2 on 8 GPUs per worker.

F.2 Corpus Construction

Both corpora are rendered from GLM-5.1-Reasoning-1M-Cleaned (Jackrong, 2026) with the target model's own chat template, so row counts are identical across model families. The `main` subset holds 328,033 rows; filtering it for fenced code blocks yields 199,704 candidate coding rows, and the `Math` topical subset holds 22,097 rows. Each domain is then reduced to exactly 20,000 rows by a deterministic scorer rather than by random sampling: every row is scored once, the top 20,000 are kept, and they are written back in source order so that row identifiers remain stable. For math20k this discards 2,097 rows, or 9.5% of the topical subset.

The score sums a length term, a domain term, and a set of hard penalties; no model is used in the loop. The length term scores prompt and answer separately against accepted bands — 40 to 4,000 characters for prompts, centered at 500, and 200 to 16,000 for answers, centered at 2,500 — with the answer term weighted more heavily, and rows outside a band are penalized in proportion to their distance from it. The domain term rewards fenced code blocks and code keywords on coding20k and LaTeX markup on math20k, together with numeric density, the presence of structural markers in the answer, and a minimum length of 80 words; each component is capped so that no single signal dominates. The penalties remove rows that are empty, that retain a reasoning-channel tag from the source rendering, that lack the assistant turn marker, that contain repeated punctuation runs or degenerate phrases, or that exceed 32,000 characters in total.

F.3 Prompting, Sampling and Answer Extraction

Generation uses vLLM with temperature 0.6, nucleus sampling at p=0.95, and a generation budget of 8,192 tokens. Each checkpoint is decoded five times from a fixed base sampling seed of 42, incremented per repeat, and the five scores are averaged to the avg@5 seed score of Appendix F.5. Answers are extracted deterministically in a fixed order: the last `\boxed{}` span, located by a brace-balanced scan so that nested expressions survive; failing that, a ####-delimited final answer; failing that, the last

number in the response. For the multiple-choice benchmarks the option order is shuffled once with seed 42 and held fixed across all conditions, so every method sees the same permutation.

Prompts are zero-shot and are assembled from four task-type templates rather than per-benchmark ones, so that every condition sees an identical instruction for a given task family. The *math* template (GSM8K, MATH-500) pairs a system turn asking for step-by-step reasoning with a user turn that restates the problem and requires the final answer inside `\boxed{}`. The *multiple-choice* template (MMLU, MMLU-STEM, MMLU-Pro, GPQA, GPQA-Diamond) asks for brief reasoning followed by a line of the form `Answer: X` naming the option letter. The *code* template (HumanEval, MBPP) asks for the complete solution in a single fenced `python` block; HumanEval supplies the function signature to complete, and MBPP supplies the task description together with the tests the function must pass. The three LiveBench tasks carry their own answer-format instructions inside the question text, so they are sent as a bare user turn with no system preamble, matching the official LiveBench protocol. All four are rendered through the target model's own chat template before decoding.

F.4 Full ID/OOD Benchmark Assignments

For math20k, OOD includes the code benchmarks (HumanEval, MBPP, LiveBench Coding) and the multi-domain block of general, STEM, professional, and reasoning QA (MMLU-STEM, MMLU, MMLU-Pro, GPQA, GPQA-Diamond, LiveBench Reasoning). For coding20k, OOD is the symmetric complement, namely the math benchmarks (GSM8K, MATH-500, LiveBench Math) plus the same multi-domain block.

The three LiveBench tasks are drawn from the `livebench/math`, `livebench/reasoning`, and `livebench/coding` test splits as retrieved in May 2026, giving 368, 200, and 128 questions respectively, with the coding split carrying LiveBench release dates 2024-06-24 and 2024-07-26. LiveBench replaces questions between releases, so scores are comparable only within a fixed snapshot, and every condition here was evaluated against the same one.

The nine-benchmark suite (Overall@9). Aggregate scores on Moonlight-16B-A3B are reported over a nine-benchmark subset of the twelve, namely GPQA-Diamond, GSM8K, HumanEval, LiveBench Coding, LiveBench Math, LiveBench Reasoning, MATH-500, MBPP, and MMLU-Pro. MMLU-STEM, MMLU, and GPQA are held out of the aggregate because they overlap heavily with MMLU-Pro and GPQA-Diamond and would repeatedly count the same knowledge axis, and they are still reported per benchmark in Table 16. Every Moonlight-16B-A3B row labeled Overall@9, **ID**, or **OOD** uses this nine-benchmark suite, with ID and OOD the split above restricted to it (math20k: 3 ID / 6 OOD, and coding20k symmetrically). Two tables report twelve-benchmark aggregates instead. The **Avg** column of Table 16 is the twelve-benchmark mean, so it does not match the Overall@9 values of Table 3; because that table lists all twelve benchmarks, averaging its nine retained columns reproduces them. The **ID**, **OOD**, and **Avg** columns of Table 12 likewise cover all twelve, but its per-benchmark companion Table 17 lists only the nine, so those columns cannot be recomputed from it.

F.5 Seed and Evaluation Aggregation Protocol

Each training seed (s42 / s43 / s44 as applicable) yields one fine-tuned checkpoint per (method, domain) cell. Each checkpoint is evaluated by `avg@5` over five sampling seeds, giving the *seed score*, and multi-seed rows report `mean`$\pm$`sd` across training seeds. Seed budgets differ between the two kinds of measurement. For *downstream performance*, the main interventional rows (RPB, LBL, SFT), the shuffled controls and Router-L2 all use three seeds (s42 / s43 / s44), while single-reference rows (Base and the hard variants) use s42 only. For *routing-graph metrics*, which require a separate probing pass over a checkpoint, RPB, LBL and SFT use three probed seeds, the shuffled controls use two (s42 / s43), and Router-L2 and the remaining single-reference conditions use one (s42). Rows that mix the two are marked in the relevant table. The main contrasts in §5.2.2 (within-soft against RPB) and Appendix A.1 (soft against hard) use within-protocol matching, so the comparisons that carry argumentative weight are between rows with matched seed budgets. Per-row counts are repeated in each table's caption.

F.6 Training Hyperparameters

All Moonlight-16B-A3B runs share the same backbone, optimizer, schedule, and batch configuration. Each corpus is trained for five epochs at a global

batch size of 128 with micro-batch 1 and a sequence length of 8,192 without packing, which is 782 optimizer steps over 20,000 rows. The optimizer is Adam (β_1=0.9, β_2=0.95) with a cosine schedule from 10^{-5} to 10^{-6}, 40 warmup steps, weight decay 0.1 and gradient clipping at 1.0. Interventions differ only in their auxiliary objective or router-attached bias: SFT runs use the cross-entropy loss alone; LBL runs add the standard load-balancing penalty at coefficient 10^{-3}, with 10^{-4} included as a robustness check; RPB runs inject the log-prior bias defined in §3.1 at the configured β and disable both the load-balancing objective and the router bias-update rule, so the prior is the only router-side signal; and Router-L2 runs add the penalty defined in §3.2 at $\lambda \in \{10^2, 10^3, 10^4\}$. DeepSeek-V2-Lite uses the analogous configuration with the same fine-tuning split.

F.7 Prior Construction (RPB)

The RPB prior tensor `pi_base[x,s,ℓ,e]` is computed in a single pass over the fine-tuning split with the frozen base model. For each sample, per-token softmaxed router probabilities at every MoE layer are aggregated over Q/A spans to a per-span scalar, yielding shape $N_{\text{samples}} \times 2 \times L \times E$, which is $20000 \times 2 \times 26 \times 64$ for Moonlight-16B-A3B. The runtime conversion to a zero-mean log-prior bias applies a per-expert clamp `clamp(p, eps/E, c/E)` (defaults $\epsilon = 0.05$, $c = 5.0$), renormalizes, takes the log, and subtracts the expert-axis mean, after clamping, so the bias has exact zero mean. The empirical per-row bias standard deviation is 0.616 on math20k and 0.654 on coding20k, with a range of ≈ 3.03, and clamp upper-bound saturation is rare, at 1–2% of cells (Table 13). β controls the overall scale, and main RPB rows use β=1.0 unless noted.

F.8 Probing Data Size

Co-activation graphs and probe statistics (sharpness, cross-domain divergence, top-k overlap to base) are computed on a held-out probing pool of $\sim$ 5,000 tokens per domain. The same tokens are used across all methods to keep metrics comparable within a (model, domain) cell. For bias-free probing (§2), the fine-tuned checkpoint is loaded without the RPB prior tensor or any training-time hook attached, and the probe pool is fed in inference mode.

F.9 Detector Configuration

Layer-wise Louvain detection (Blondel et al., 2008) uses the standard modularity objective Q (Newman, 2006) with resolution parameter $\gamma = 1.0$ and a maximum of 100 iterations or convergence within $\Delta Q < 10^{-6}$. The input to Louvain is the symmetric, non-negative, zero-diagonal co-activation matrix $W^{(\ell)}$ per layer, after the edge-fraction threshold of §2 retains the top 10% of edges by weight, with sweeps over 0.05, 0.15 and 0.20 in Appendix B.2. Edge weights are not row-normalized, since row-normalization changes the modularity geometry, so we use the raw co-occurrence frequencies. Leiden (Traag et al., 2018) is used in the robustness sweep at otherwise-matched parameters. Detector seeds 0, 1, 2 and 42 are run in the appendix, while the main-paper rows use seed 0.

F.10 Community NMI Aggregation

Per-layer NMI is computed with the standard symmetric normalization, $2I(X;Y)/(H(X)+H(Y))$. The scalar community NMI we report per checkpoint is the unweighted *layer mean* across all MoE layers, and Q and ΔQ follow the same aggregation.

G Full Result Tables

G.1 Routing-Graph Metrics

Table 15 reports the routing-state measurements behind the retention hierarchy of §5.2.1, together with downstream performance, for SFT, LBL, RPB and Router-L2. Community NMI separates the two soft anchors from SFT and LBL on both splits, while TVD and JSD place all four methods in the same range and top-k overlap leaves SFT between the two anchors on coding20k.

G.2 Per-Benchmark Main Interventions

Table 16 breaks the Moonlight-16B-A3B headline results (§5.1.1) into individual benchmarks, adding the base checkpoint as a reference row, and groups the benchmarks as multi-domain, code, and math. Table 17 gives the same breakdown for the four main interventions on DeepSeek-V2-Lite at a single training seed (s42). The DeepSeek-V2-Lite table is included for completeness of the scope condition of §5.2.3 rather than as an independent multi-seed claim.

G.3 Per-Benchmark Shuffled-Prior Controls

Table 5 in §5.2.2 reports the soft and hard dissociation at the aggregate level, as one Overall@9

Dataset	Method	Routing-Graph Metrics				Performance	
		Comm. NMI	TVD	JSD	top-k	ID	OOD
math20k	SFT	$0.575_{\pm 0.004}$	$0.320_{\pm 0.000}$	$0.093_{\pm 0.000}$	$0.469_{\pm 0.001}$	$29.44_{\pm 0.32}$	$15.65_{\pm 0.38}$
	LBL	$0.554_{\pm 0.006}$	$0.306_{\pm 0.001}$	$0.082_{\pm 0.000}$	$0.420_{\pm 0.002}$	$31.91_{\pm 0.10}$	$14.97_{\pm 0.51}$
	RPB	$\mathbf{0.631}_{\pm 0.003}$	$0.310_{\pm 0.001}$	$0.089_{\pm 0.000}$	$\mathbf{0.485}_{\pm 0.001}$	$\mathbf{45.77}_{\pm 0.84}$	$\mathbf{19.53}_{\pm 0.28}$
	Router-L2†	0.628	0.312	0.090	0.480	$44.38_{\pm 0.21}$	$19.34_{\pm 0.33}$
coding20k	SFT	$0.588_{\pm 0.001}$	$0.315_{\pm 0.001}$	$0.084_{\pm 0.001}$	$0.482_{\pm 0.002}$	$28.34_{\pm 1.36}$	$35.83_{\pm 0.49}$
	LBL	$0.544_{\pm 0.003}$	$0.317_{\pm 0.001}$	$0.081_{\pm 0.000}$	$0.391_{\pm 0.002}$	$30.73_{\pm 0.13}$	$35.56_{\pm 0.38}$
	RPB	$0.629_{\pm 0.001}$	$0.327_{\pm 0.001}$	$0.088_{\pm 0.000}$	$0.481_{\pm 0.001}$	$41.37_{\pm 0.30}$	$\mathbf{38.70}_{\pm 0.11}$
	Router-L2†	**0.632**	0.321	0.086	**0.486**	$\mathbf{41.50}_{\pm 0.52}$	$38.28_{\pm 0.37}$

Table 15: Routing-graph metrics and downstream performance for SFT, LBL, RPB, and Router-L2 (λ=10^4) on math20k and coding20k, all on Moonlight-16B-A3B. **ID** and **OOD** are seed-level means over the nine retained benchmarks (Appendix F.4). For SFT, LBL and RPB every column is a three-seed mean $\pm$ sample standard deviation (s42 / s43 / s44), with each performance seed score itself an avg@5 over five evaluation repeats. For Router-L2 (†) only the performance columns are three-seed means, since its routing-graph metrics come from a single probed checkpoint (s42) and are reported without a spread. Higher community NMI and top-k overlap indicate retention of base co-activation structure, while TVD and JSD measure marginal-distribution shift. Bold marks the best mean within each dataset block on community NMI, top-k overlap, ID, and OOD, and TVD and JSD are left unbolded because neither direction is unambiguously better.

figure and one community NMI per condition. The per-benchmark breakdown is given here together with the matched unshuffled RPB row, so the ordering can be checked column by column rather than only at the aggregate level. On Moonlight-16B-A3B the ordering is RPB $\approx$ within-soft $\gg$ within-hard $>$ global-hard on both fine-tuning splits (Table 18), over the three training seeds used elsewhere in the paper. On DeepSeek-V2-Lite the soft against hard gap reproduces, while global-hard sits slightly above within-hard on both splits (Table 19), so the monotone dependence on shuffle scope is a Moonlight-16B-A3B result rather than a general one. The DeepSeek-V2-Lite table is at a single training seed, s42 throughout except coding20k within-soft, which falls back to s43, and the spread reported per cell is the within-seed across-rep standard deviation (avg@5 with sample sd). The RPB rows of both tables are the reference values used to read the soft against hard dissociation in §5.2.2, where within-soft sits within seed noise of RPB on Moonlight-16B-A3B while the hard variants fall away from it.

Method	Dataset	Multi-Domain						Code			Math			Overall
		MMLU-STEM	MMLU	MMLU-Pro	GPQA	GPQA-Diamond	LiveBench Reasoning	HumanEval	MBPP	LiveBench Code	GSM8K	MATH-500	LiveBench Math	Avg
none	Base	$26.26_{\pm 0.33}$	$26.43_{\pm 1.00}$	$13.64_{\pm 0.76}$	$24.55_{\pm 1.10}$	$21.11_{\pm 2.84}$	$4.00_{\pm 2.55}$	$16.46_{\pm 7.07}$	$14.08_{\pm 7.67}$	$6.09_{\pm 4.07}$	$44.56_{\pm 19.96}$	$26.24_{\pm 10.08}$	$9.76_{\pm 3.26}$	$19.43_{\pm 3.79}$
SFT	math	$15.80_{\pm 0.44}$	$13.71_{\pm 0.53}$	$14.26_{\pm 0.30}$	$10.28_{\pm 0.52}$	$9.33_{\pm 0.51}$	$9.73_{\pm 1.50}$	$24.72_{\pm 2.17}$	$24.19_{\pm 0.58}$	$11.67_{\pm 1.06}$	$59.57_{\pm 0.47}$	$19.51_{\pm 0.30}$	$9.24_{\pm 0.61}$	$18.50_{\pm 0.23}$
LBL	math	$15.17_{\pm 1.38}$	$12.27_{\pm 1.83}$	$13.58_{\pm 0.33}$	$9.54_{\pm 1.06}$	$8.92_{\pm 1.13}$	$8.93_{\pm 0.45}$	$23.41_{\pm 1.06}$	$23.29_{\pm 2.45}$	$11.67_{\pm 1.40}$	$63.85_{\pm 0.32}$	$22.08_{\pm 0.45}$	$9.79_{\pm 0.06}$	$18.54_{\pm 0.58}$
RPB	math	$\mathbf{19.16}_{\pm 0.75}$	$\mathbf{16.27}_{\pm 0.95}$	$\mathbf{18.59}_{\pm 0.79}$	$\mathbf{11.49}_{\pm 0.84}$	$\mathbf{11.41}_{\pm 1.11}$	$\mathbf{11.37}_{\pm 0.46}$	$\mathbf{33.01}_{\pm 1.83}$	$\mathbf{26.57}_{\pm 4.20}$	$\mathbf{16.20}_{\pm 3.36}$	$\mathbf{76.38}_{\pm 0.63}$	$\mathbf{41.88}_{\pm 1.33}$	$\mathbf{19.05}_{\pm 1.37}$	$\mathbf{25.11}_{\pm 0.03}$
SFT	code	$56.51_{\pm 0.41}$	$\mathbf{56.31}_{\pm 0.62}$	$35.80_{\pm 0.42}$	$28.17_{\pm 0.54}$	$27.74_{\pm 1.47}$	$\mathbf{13.83}_{\pm 1.78}$	$43.58_{\pm 2.24}$	$30.88_{\pm 2.03}$	$10.57_{\pm 0.09}$	$75.38_{\pm 0.42}$	$43.29_{\pm 1.44}$	$18.93_{\pm 0.64}$	$36.75_{\pm 0.48}$
LBL	code	$51.79_{\pm 1.43}$	$49.91_{\pm 1.46}$	$33.66_{\pm 0.27}$	$27.31_{\pm 0.75}$	$\mathbf{28.69}_{\pm 0.52}$	$12.27_{\pm 1.68}$	$46.46_{\pm 1.28}$	$30.11_{\pm 0.88}$	$15.62_{\pm 0.98}$	$76.85_{\pm 0.33}$	$42.84_{\pm 0.22}$	$19.07_{\pm 0.24}$	$36.21_{\pm 0.36}$
RPB	code	$\mathbf{57.89}_{\pm 1.81}$	$55.21_{\pm 2.70}$	$\mathbf{37.81}_{\pm 1.03}$	$\mathbf{28.96}_{\pm 0.57}$	$27.88_{\pm 1.49}$	$13.33_{\pm 0.60}$	$\mathbf{66.54}_{\pm 0.74}$	$\mathbf{37.63}_{\pm 1.10}$	$\mathbf{19.95}_{\pm 0.39}$	$\mathbf{81.16}_{\pm 0.47}$	$\mathbf{50.48}_{\pm 0.33}$	$\mathbf{21.56}_{\pm 0.43}$	$\mathbf{41.53}_{\pm 0.35}$

Table 16: Moonlight-16B-A3B per-benchmark results grouped by fine-tuning dataset and benchmark family. For the three fine-tuning methods, each cell is the mean ± sample standard deviation across three training seeds (s42 / s43 / s44), with each seed score itself an avg@5 over five evaluation repeats, matching the seed protocol of Table 2. The **Avg** column, unlike that table, is the mean over all twelve benchmarks, and averaging the nine retained columns instead reproduces the Overall@9 values of Table 3. The **Base** row is the pretrained checkpoint before fine-tuning, which has no training seed, so its spread is over the five evaluation repeats. Bold indicates the best mean per benchmark column within each fine-tuning-dataset block.

Dataset	Method	Multi-Domain			Math			Code			Overall
		MMLU-Pro	GPQA-Diamond	LiveBench Reasoning	GSM8K	MATH-500	LiveBench Math	HumanEval	MBPP	LiveBench Code	Avg
none	Base	$13.96_{\pm 1.27}$	$17.47_{\pm 5.43}$	$2.20_{\pm 0.57}$	$4.40_{\pm 0.78}$	$3.92_{\pm 1.66}$	$2.99_{\pm 0.76}$	$22.07_{\pm 3.21}$	$27.24_{\pm 5.06}$	$1.72_{\pm 1.40}$	10.66
math20k	SFT	$14.29_{\pm 3.27}$	$6.77_{\pm 2.44}$	$7.50_{\pm 2.06}$	$18.98_{\pm 2.25}$	$11.92_{\pm 1.15}$	$\mathbf{9.76}_{\pm 0.97}$	$29.51_{\pm 4.73}$	$29.48_{\pm 3.07}$	$6.56_{\pm 2.51}$	14.98
math20k	LBL	$15.25_{\pm 3.44}$	$7.17_{\pm 2.87}$	$\mathbf{9.10}_{\pm 1.39}$	$19.26_{\pm 1.60}$	$11.28_{\pm 1.05}$	$9.74_{\pm 1.03}$	$\mathbf{32.07}_{\pm 3.24}$	$30.64_{\pm 3.03}$	$\mathbf{7.50}_{\pm 1.42}$	15.78
math20k	RPB	$\mathbf{17.13}_{\pm 3.05}$	$\mathbf{10.20}_{\pm 4.76}$	$7.90_{\pm 2.95}$	$\mathbf{20.94}_{\pm 0.66}$	$\mathbf{12.08}_{\pm 0.92}$	$9.30_{\pm 1.40}$	$30.37_{\pm 4.34}$	$\mathbf{32.28}_{\pm 2.41}$	$6.41_{\pm 1.28}$	**16.29**
coding20k	SFT	$19.53_{\pm 0.95}$	$\mathbf{22.83}_{\pm 3.97}$	$8.40_{\pm 2.97}$	$\mathbf{25.53}_{\pm 1.14}$	$\mathbf{11.24}_{\pm 1.09}$	$8.14_{\pm 1.74}$	$31.83_{\pm 3.00}$	$26.44_{\pm 2.62}$	$\mathbf{7.50}_{\pm 1.52}$	17.94
coding20k	LBL	$18.31_{\pm 0.73}$	$21.72_{\pm 5.70}$	$\mathbf{10.20}_{\pm 1.57}$	$24.90_{\pm 1.01}$	$11.16_{\pm 1.40}$	$\mathbf{8.40}_{\pm 1.19}$	$31.95_{\pm 3.16}$	$27.08_{\pm 2.42}$	$6.56_{\pm 1.62}$	17.81
coding20k	RPB	$\mathbf{21.34}_{\pm 0.75}$	$21.82_{\pm 4.51}$	$8.60_{\pm 1.64}$	$22.90_{\pm 1.40}$	$11.12_{\pm 1.14}$	$7.54_{\pm 0.70}$	$\mathbf{32.93}_{\pm 2.89}$	$\mathbf{28.20}_{\pm 0.76}$	$\mathbf{7.50}_{\pm 2.63}$	**17.99**

Table 17: DeepSeek-V2-Lite per-benchmark results for the four main interventions (Base / SFT / LBL / RPB) on math20k and coding20k. Each cell is mean ± standard deviation over five evaluation repeats (avg@5) at a single training seed (s42). **Avg** aggregates by averaging the nine retained benchmark columns. Bold indicates the best mean within each fine-tuning dataset block per benchmark column, and ties on the displayed precision are both bolded.

Dataset	Method	Multi-Domain			Math			Code			Overall
		MMLU-Pro	GPQA-Diamond	LiveBench Reasoning	GSM8K	MATH-500	LiveBench Math	HumanEval	MBPP	LiveBench Code	Avg
math20k	RPB	$18.59_{\pm 0.79}$	$11.41_{\pm 1.11}$	$11.37_{\pm 0.46}$	$76.38_{\pm 0.63}$	$41.88_{\pm 1.33}$	$19.05_{\pm 1.37}$	$33.01_{\pm 1.83}$	$26.57_{\pm 4.20}$	$16.20_{\pm 3.36}$	28.27
	within-soft	$18.50_{\pm 1.52}$	$11.11_{\pm 1.67}$	$10.50_{\pm 0.44}$	$75.75_{\pm 1.59}$	$42.68_{\pm 0.62}$	$19.53_{\pm 0.43}$	$30.73_{\pm 0.68}$	$19.99_{\pm 1.06}$	$16.30_{\pm 1.58}$	27.23
	within-hard	$10.65_{\pm 0.43}$	$3.50_{\pm 0.56}$	$8.80_{\pm 0.44}$	$53.97_{\pm 2.57}$	$26.89_{\pm 2.17}$	$11.23_{\pm 0.44}$	$21.99_{\pm 9.45}$	$26.12_{\pm 1.18}$	$5.78_{\pm 2.34}$	18.77
	global-hard	$9.53_{\pm 0.37}$	$3.50_{\pm 3.54}$	$6.93_{\pm 0.67}$	$3.36_{\pm 1.53}$	$3.83_{\pm 0.12}$	$4.73_{\pm 1.44}$	$0.04_{\pm 0.07}$	$0.16_{\pm 0.18}$	$0.21_{\pm 0.09}$	3.59
coding20k	RPB	$37.81_{\pm 1.03}$	$27.88_{\pm 1.49}$	$13.33_{\pm 0.60}$	$81.16_{\pm 0.47}$	$50.48_{\pm 0.33}$	$21.56_{\pm 0.43}$	$66.54_{\pm 0.74}$	$37.63_{\pm 1.10}$	$19.95_{\pm 0.39}$	39.59
	within-soft	$35.43_{\pm 0.75}$	$27.58_{\pm 0.44}$	$11.90_{\pm 0.36}$	$80.83_{\pm 0.53}$	$49.65_{\pm 0.19}$	$21.13_{\pm 0.88}$	$63.98_{\pm 3.14}$	$41.53_{\pm 3.76}$	$18.65_{\pm 0.80}$	38.96
	within-hard	$18.79_{\pm 2.21}$	$20.37_{\pm 1.02}$	$11.63_{\pm 0.31}$	$60.49_{\pm 1.65}$	$30.00_{\pm 1.29}$	$13.12_{\pm 0.78}$	$34.39_{\pm 2.66}$	$24.72_{\pm 2.02}$	$10.31_{\pm 0.87}$	24.87
	global-hard	$16.08_{\pm 0.49}$	$23.00_{\pm 0.65}$	$10.23_{\pm 1.44}$	$45.50_{\pm 3.95}$	$20.01_{\pm 2.48}$	$8.07_{\pm 0.85}$	$18.66_{\pm 7.82}$	$15.48_{\pm 5.38}$	$4.58_{\pm 2.96}$	17.96

Table 18: Moonlight-16B-A3B per-benchmark results for the soft-anchoring reference (RPB) and shuffled-prior controls (within-soft, within-hard, global-hard) on math20k and coding20k. Each cell is mean ± standard deviation across three training seeds (s42 / s43 / s44), where each seed score is itself avg@5 over five evaluation repeats. **Avg** aggregates by averaging the nine retained benchmark columns.

Dataset	Method	Multi-Domain			Math			Code			Overall
		MMLU-Pro	GPQA-Diamond	LiveBench Reasoning	GSM8K	MATH-500	LiveBench Math	HumanEval	MBPP	LiveBench Code	Avg
math20k	RPB	$17.13_{\pm 3.05}$	$10.20_{\pm 4.76}$	$7.90_{\pm 2.95}$	$20.94_{\pm 0.66}$	$12.08_{\pm 0.92}$	$9.30_{\pm 1.40}$	$30.37_{\pm 4.34}$	$32.28_{\pm 2.41}$	$6.41_{\pm 1.28}$	16.29
	within-soft	$16.30_{\pm 3.23}$	$6.97_{\pm 4.39}$	$9.10_{\pm 1.52}$	$21.85_{\pm 1.35}$	$12.00_{\pm 1.46}$	$9.92_{\pm 1.91}$	$27.32_{\pm 3.94}$	$32.76_{\pm 1.64}$	$6.88_{\pm 2.56}$	15.90
	within-hard	$10.15_{\pm 0.23}$	$1.82_{\pm 0.77}$	$6.00_{\pm 1.37}$	$12.83_{\pm 1.32}$	$7.64_{\pm 0.95}$	$8.48_{\pm 1.40}$	$0.98_{\pm 0.92}$	$2.12_{\pm 0.73}$	$0.16_{\pm 0.35}$	5.58
	global-hard	$10.15_{\pm 0.13}$	$2.32_{\pm 1.05}$	$7.40_{\pm 2.82}$	$12.57_{\pm 1.27}$	$7.20_{\pm 0.75}$	$8.57_{\pm 1.05}$	$7.20_{\pm 1.46}$	$13.56_{\pm 1.53}$	$1.25_{\pm 0.89}$	7.80
coding20k	RPB	$21.34_{\pm 0.75}$	$21.82_{\pm 4.51}$	$8.60_{\pm 1.64}$	$22.90_{\pm 1.40}$	$11.12_{\pm 1.14}$	$7.54_{\pm 0.70}$	$32.93_{\pm 2.89}$	$28.20_{\pm 0.76}$	$7.50_{\pm 2.63}$	17.99
	within-soft	$21.80_{\pm 0.42}$	$22.22_{\pm 1.18}$	$8.50_{\pm 2.18}$	$23.28_{\pm 0.52}$	$11.80_{\pm 1.02}$	$7.89_{\pm 1.01}$	$32.93_{\pm 3.48}$	$29.36_{\pm 1.95}$	$8.28_{\pm 2.45}$	18.45
	within-hard	$14.10_{\pm 0.42}$	$20.20_{\pm 2.20}$	$7.00_{\pm 1.00}$	$9.16_{\pm 0.53}$	$5.56_{\pm 0.99}$	$5.89_{\pm 0.95}$	$4.63_{\pm 1.02}$	$7.56_{\pm 0.85}$	$0.62_{\pm 0.35}$	8.30
	global-hard	$13.67_{\pm 0.53}$	$21.92_{\pm 4.84}$	$7.40_{\pm 1.29}$	$10.52_{\pm 1.29}$	$4.72_{\pm 1.43}$	$5.32_{\pm 0.85}$	$7.32_{\pm 3.14}$	$9.04_{\pm 1.77}$	$1.56_{\pm 1.10}$	9.05

Table 19: DeepSeek-V2-Lite per-benchmark results for the soft-anchoring reference (RPB) and shuffled-prior controls (within-soft, within-hard, global-hard) on math20k and coding20k. Each cell is mean ± standard deviation over five evaluation repeats (avg@5) at a single training seed (s42 where available, and coding20k within-soft uses s43 because s42 was not trained). **Avg** aggregates by averaging the nine retained benchmark columns.