BEYOND LOAD BALANCING: EXPERT ASSIGNMENT MATTERS IN GENOMIC MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Genomic mixture-of-experts (MoE) models inherit load-balancing objectives from language modelling. We ask what these objectives control. In a released JanusDNA checkpoint, the shipped objective is nearly satisfied only after routers are pooled across layers (pooled Gini 0.0068; individual layers 0.164–0.407). Adding a per-layer term flattens every layer, and retargeting it to a genome-composition prior moves expert utilisation up to a 5.06–$5.98\times$ within-layer max/min ratio, while validation loss across fourteen non-divergent fine-tuned checkpoints spans only 0.06%, and generation coverage and three linear probes show no attributable gain. By contrast, permuting expert identity at inference, which leaves the utilisation histogram unchanged up to relabelling, raises validation loss by 7.48% on the same 128 evaluation windows. In a separate decomposition, recomputing gate weights retains 89% of the permutation cost, and swapping a single expert pair costs about as much per rerouted slot as a full permutation. Permutation is more damaging than the most consequential single-expert ablation in two further genomic MoEs spanning a $34\times$ range in scale. The objective is not inert: arms with a per-layer prior-targeted term show about 39% lower layer-7 permutation sensitivity, a separation that holds across three seeds of two arms, although we do not establish whether this reflects harm or redundancy. We therefore recommend monitoring permutation sensitivity, class enrichment against a matched null, and expert-output effective rank alongside utilisation, rather than replacing balancing with an unvalidated objective.

1 INTRODUCTION

Most of the human genome is non-coding, and many variants associated with complex traits and disease lie outside protein-coding regions. Reading what a stretch of DNA does therefore means reading a regulatory grammar that is written in the sequence but not directly legible from it, and genomic language models are increasingly used for variant interpretation, regulatory annotation and sequence design (e.g., Brixi et al., 2026); we review them in §2. The setting is demanding: dependencies span hundreds of kilobases, nearly half the genome is repetitive, and one four-letter alphabet describes exons, promoters, enhancers and transposable elements with markedly different statistics.

Mixture-of-experts (MoE) models offer a natural way to handle these regimes: a router sends each piece of sequence to one or more expert sub-networks (e.g., Duan et al., 2025). Whether different regimes reach different experts depends on what the router learns, and in the models we study, routing is regularised by load-balancing objectives inherited from language modelling. This paper asks what those objectives actually control.

A router fixes two things at once (Figure 1): *utilisation*, how much traffic each expert receives, which the balancing objective constrains; and *assignment*, which token goes to which expert, which it does not directly specify. We intervene on each, and the two intervention families answer different questions: changing the objective during fine-tuning asks what a balancing term can move, while perturbing assignments at inference on fixed weights asks what an already trained model relies on. A per-layer, composition-targeted objective moves utilisation up to a 5.06–$5.98\times$ within-layer max/min ratio, while validation loss across fourteen non-divergent fine-tuned checkpoints spans only 0.06% (§4). On the same 128 windows, inference-time expert permutation raises validation loss by 7.48% without changing utilisation up to relabelling (§5). The two numbers come from different kinds of

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.

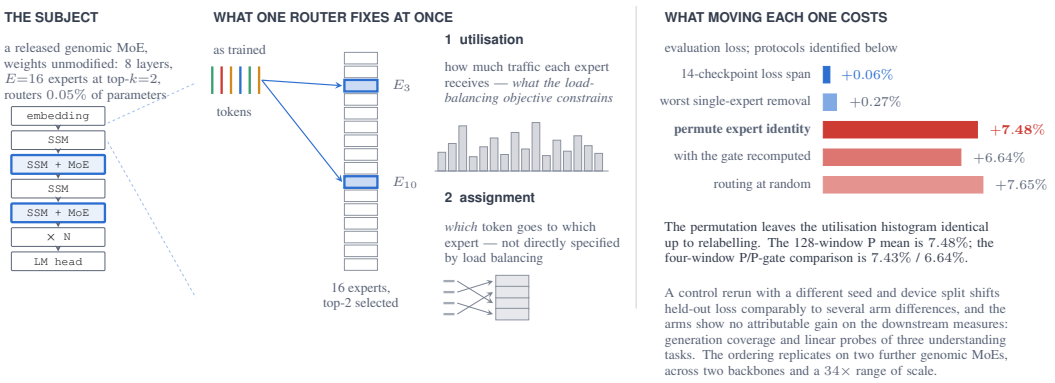

Figure 1: **What this paper measures.** One router fixes two things at once, and the studied load-balancing objective directly regulates the first. The 6× utilisation ratio belongs to A4nb0 within layers, while 0.06% is the validation-loss span across trained JanusDNA arms on 128 windows. Inference-time all-layer permutation on A0 incurs 7.48% on the same 128 windows, about 126× that span (a descriptive ratio, not a causal effect estimate). Other plotted perturbations use the separate four-window decomposition: its P mean is 7.43%; P-gate, random routing, and worst ablation are 6.64%, 7.65%, and 0.27%, respectively.

intervention, so their ratio is descriptive rather than a causal effect estimate; their order is the finding. The objective regulates utilisation; validation loss is far more sensitive to assignment.

That was not the result we set out to find. We reached it through two interventions on the objective, each of which works on its target but neither of which yields an attributable downstream gain. The first changes *where* the objective is computed: the released implementation pools router logits across layers, where layer-specific skews cancel. The second changes *what* it targets: uniform utilisation is not an obvious goal for a genome in which nearly half the sequence is repetitive, so we aim the objective at a composition-derived prior. Expert load, in other words, can move substantially while downstream behaviour stays put; §5 asks why.

Our contribution is diagnostic rather than a new state-of-the-art genomic predictor: (i) showing that the shipped objective is near-satisfied only after pooling across layers, so layer-wise imbalance is invisible to it (§4); (ii) decomposing the permutation cost to show that it arises mainly from token–expert correspondence rather than gate-weight mismatch, and that arms with a per-layer prior-targeted term show about 39% lower layer-7 permutation sensitivity, a separation that holds across three seeds (§5); (iii) replicating the permutation-over-ablation ordering in PlantBiMoE and Genos-1.2B (§6); and (iv) proposing three structural statistics to monitor alongside utilisation (§8). Held-out differences among fine-tuned checkpoints are separable but not attributable with our seed evidence (§4.2), and we establish neither downstream benefit nor harm. Secondary results—an exploratory reverse-complement constraint (§7), an unsuccessful block-level router, and a reconstruction-score-based generator-selection failure (§4.1)—are reported with a pre-registration; overturned claims are recorded in the supplement.

2 Background

Genomic sequence models. DNABERT (Ji et al., 2021) and Nucleotide Transformer (Dalla-Torre et al., 2025) established pretraining directly on DNA; HyenaDNA (Nguyen et al., 2023) and Caduceus (Schiff et al., 2024) extended context with long implicit convolutions and state-space layers, the latter building in reverse-complement equivariance. Evo 2 (Brixi et al., 2026) generates DNA at genomic scale, and AlphaGenome (Avsec et al., 2026) maps up to 1 Mb of sequence to functional genomic tracks. Realistic local statistics in generated DNA do not guarantee realistic long-range organisation (Tzanakakis et al., 2026); we return to this evaluation problem in §4.1.

Load balancing. The sparsely-gated MoE introduced an auxiliary load-balancing loss to discourage routing collapse (Shazeer et al., 2017). GShard (Lepikhin et al., 2021), Switch (Fedus et al., 2022)

and ST-MoE (Zoph et al., 2022) keep a scalar penalty on expert utilisation, and later work changes how balance is enforced, through expert-choice routing (Zhou et al., 2022) or a bias term without an auxiliary loss (Wang et al., 2024). These methods do not ask what model behaviour is lost or preserved when the utilisation histogram itself moves. Genomic MoEs adapt such routing to DNA: JanusDNA (Duan et al., 2025) is a bidirectional hybrid foundation model, PlantBiMoE (Lin et al., 2025b) a bidirectional sparse-MoE model for plant genomes, SPACE (Yang et al., 2025) a profile-predicting DNA model with expert routing, and Genos-1.2B (Lin et al., 2025a) a human-centric foundation model. All four regulate expert load; none, to our knowledge, reports what the routing assignment itself carries. In DNA, a composition-derived target is a prior on expert traffic, not a supervision signal saying which functional class should reach which expert, and a dead expert is a capacity failure however class enrichment looks.

What routers encode. Evidence from language MoEs is mixed and mostly observational: Mixtral routes by syntax more than by topic (Jiang et al., 2024), ST-MoE finds lexical specialisation mainly in encoder experts (Zoph et al., 2022), routing can cluster representations around expert centroids (Chi et al., 2022), and fine-grained and shared experts reshape how specialisation is distributed (Dai et al., 2024). We instead intervene directly on the learned assignment.

3 Experimental setup

Models. Our primary model is the released JanusDNA checkpoint with weights unchanged (Harvard Dataverse, `doi:10.7910/DVN/HDT0RN`; md5 verified): 35.6 M parameters, eight Mamba layers with MoE at layers 1, 3, 5 and 7 and a final attention block (attention and a dense MLP, no experts), 16 experts with top-2 routing, eight unshared routers holding 0.05% of the parameters, and a 131,072 bp context. It is a bidirectional denoiser: input and target are aligned without a shift (`data == target`), no positions are masked, and cross-entropy is computed over all positions. Our validation losses are therefore unmasked reconstruction-style scores, not held-out masked-token or autoregressive likelihoods. The A0 baseline is about 0.8945 nats, unlike the near-zero 0.0047 nats of an unmasked PlantBiMoE evaluation (Appendix I). At the fusion step, the prediction at position t combines the forward state at $t-1$ and the re-reversed backward state at $t+1$, rather than the unshifted states at t (the two boundary positions use a single stream). These neighbouring states and the terminal attention block may nevertheless carry information about the original token, so the score is a reconstruction-style quantity rather than a strictly held-out prediction. The iterative Gibbs sampler follows a different path (Appendix D). Bidirectionality concatenates the forward embeddings with their reversal, which provides some reversal symmetry but no complement symmetry; reverse-complement augmentation is disabled. We repeat the key inference-time interventions on PlantBiMoE (Lin et al., 2025b) (140.8 M parameters, four experts with top-1 routing, taken from the checkpoint configuration rather than the class defaults) and Genos-1.2B (Lin et al., 2025a) (eight experts, top-2, autoregressive Transformer); their evaluation protocols are in Appendix I.

Arms and protocol. Every constraint attaches through forward hooks or training callbacks, so the released model code stays byte-identical. Each JanusDNA arm fine-tunes from the released checkpoint with seed 2222, learning rate 8×10^{-4}, global batch 8 and 10,000 steps; arms differ in their routing objectives or in explicitly specified router-input and auxiliary-head interventions. The main arms form a cumulative ladder. A0 is the control with the shipped pooled objective (coefficient 0.2), and A0z sets that coefficient to zero. A1 adds a reverse-complement routing term ($\lambda = 0.05$; A1w is a depth-weighted variant); A2 adds a smoothness term (10^{-3}) to A1; A4nb adds the per-layer prior-targeted term ($\lambda = 0.0019$) to A2; A4nb0 is A4nb with the shipped pooled term switched off; and A5nb adds multi-token-prediction heads to A4nb0. A3 and A3c route from pooled blocks, and A6 and A6h add an attribution-based assignment term. The full grid is in Appendix A. Logged validation loss depends on the `(devices, accumulation)` configuration (Appendix A), so every reported validation-loss value is recomputed offline from the step-10,000 checkpoints on the same 128 held-out windows in the same order; the reference A0 run scores 0.894469 nats. The fourteen non-divergent checkpoint evaluations span 0.06%; an A0 rerun with a different seed and device/accumulation configuration lies 0.00033 nats above the reference, against a total range of 0.00053 nats (Appendix A); a divergent block-routing arm (A3c) is kept as a positive control.

Pre-registration. Thresholds and selection rules were fixed before the runs they governed, and we report outcomes that did not support our expectations: one unsupported prediction (§4) and one non-discriminating selection rule (§7). The learning rate is the largest candidate with zero dead experts through step 3,000; it selects 8×10^{-4} despite the worst validation loss of the three candidates. One abort criterion for block-level routing was added post hoc and is marked as such (Appendix E).

Measures. *Permutation sensitivity* is the relative increase in validation loss when expert identities are permuted at inference, with weights fixed and the utilisation histogram preserved up to relabelling; it isolates dependence on the learned assignment from the histogram and is an operational measure, not an information-theoretic quantity. We report it for a single layer (layer-ℓ) or for all four MoE layers together (P, original gate weights retained). The headline P uses the 128 windows above (mean $\pm$ SD over 10 permutations); a separate decomposition on four 131,072 bp windows (five permutations) compares P with gate recomputation, random routing and ablation. *Class enrichment* is the strongest per-expert $|\log_2|$ enrichment of an annotation class, compared against a matched null that permutes token–expert assignments within each window, preserving class composition and each expert's utilisation. *Effective rank* is the singular-value participation ratio of all 16 experts' outputs on 2,048 shared token representations. Routing agreement under a sequence transformation is a chance-corrected overlap $\kappa = (o - n)/(1 - n)$, reported against both a uniform null and a matched null that preserves marginal usage, because the uniform null alone is misleading when usage is non-uniform (Appendix I). Utilisation Gini uses four windows for the released checkpoint and eight for trained arms, and is reported with dead-expert counts, since entropy alone can miss concentration: at the released checkpoint, layer 7 has normalised entropy 0.937 yet block-pooled Gini 0.696 with no dead expert (Appendix I).

4 MOVING EXPERT LOAD HAS LITTLE ATTRIBUTABLE LOSS COST

The objective is near-satisfied only after pooling. The released implementation's `load_balancing_loss_func` concatenates router logits across layers before computing one balance term. At the released checkpoint, token-level utilisation has a pooled Gini of only 0.0068, while the four MoE layers read 0.270, 0.164, 0.262 and 0.407 on the same four 131 kbp windows (Figure 2a): layers favour different experts, and their skews cancel when pooled, leaving the pooled value 24–$60\times$ smaller than the layer-wise ones. A coefficient that is simply too small does not explain this. At identical routing, the per-layer form carries a router gradient $41\times$ the pooled one; the released term already supplies 36% of the language-modelling gradient at the router; and in a pre-registered coefficient sweep, raising the coefficient from 0.2 to 2.0 increased layer-7 Gini from 0.454 to 0.516 and mean per-layer Gini from 0.290 to 0.321, contrary to the registered null prediction (Appendix J). The problem is the level at which balance is measured, not how strongly it is weighted.

We therefore compute the standard objective per layer, $\mathcal{L}_{\text{bal}} = \frac{1}{K}\sum_{k=1}^{K}\sum_{e=1}^{E} f_{k,e}P_{k,e}/t_e$, where $f_{k,e}$ is the fraction of tokens routed to expert e in MoE layer k, $P_{k,e}$ its mean gate probability and t_e its target share. Setting $t_e = 1/E$ recovers the uniform objective, which we verify numerically; the composition-targeted variant sets t_e from a composition-derived prior π. Because per-layer forms carry much larger router gradients, coefficients are calibrated to the released term's gradient share ($\lambda = 0.0019$ for the prior-targeted form used by A4nb and A4nb0; Appendix J): the same numerical coefficient does not exert the same optimisation force once the objective is computed per layer or retargeted. Genos-1.2B independently adds a balanced-load mechanism for imbalance its authors associate with the four-base vocabulary (Lin et al., 2025a); our intervention instead targets how the objective is aggregated across layers.

Where. A per-layer term exposes the imbalance that the pooled term cannot see. In A4nb, which applies the objective per layer with the shipped uniform-targeted term still running, against A0 token-level Gini at layers 1, 3, 5 and 7 falls from 0.221, 0.104, 0.194 and 0.334 to 0.032, 0.029, 0.031 and 0.038 (3.6–$8.8\times$; eight windows; Figure 2b), at $+0.05\%$ validation loss. This flattening reflects both terms together: with the shipped term off, A4nb0 settles at 0.134–0.153 (see What). A4nb also carries A2's reverse-complement and smoothness terms, so this contrast with A0 is cumulative; §5 instead contrasts A4nb with A2. Strengthening the pooled term alone made the same layer-wise imbalance worse, so the aggregation level, not the coefficient, determines what the objective changes. The loss difference is small but not zero; its attribution is tested in §4.2.

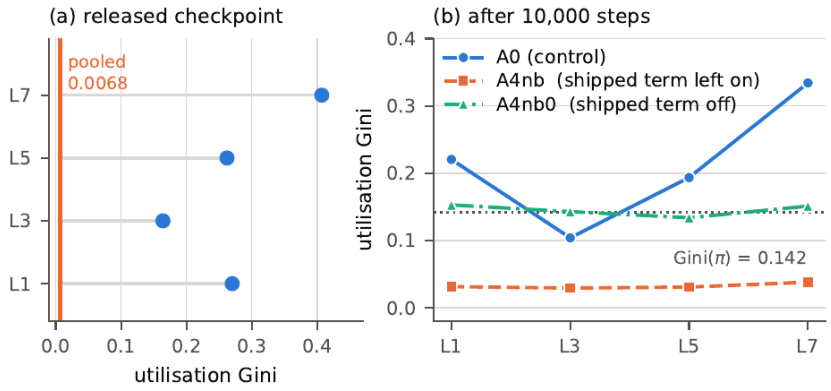

Figure 2: The balancing objective optimises a statistic that is already satisfied. **(a)** At the released checkpoint the pooled utilisation Gini corresponding to the shipped across-layer aggregation is 0.0068, while the four MoE layers it pools over run 0.164–0.407. **(b)** Per-layer Gini after 10,000 fine-tuning steps (eight windows). A4nb and A4nb0 aim the same per-layer objective at the prior π and differ only in whether the shipped uniform-targeted term was left running: with it on, utilisation flattens toward uniform; with it off, it settles at the prior's own inequality, $\mathrm{Gini}(\pi) = 0.142$. Flatness is the right summary only while the target is uniform.

What. Gini cannot distinguish matching a skewed target from flattening toward uniform. A4nb's per-layer term targets the prior π, yet with the shipped uniform term competing, it flattens: its mean L_1 distance is 0.20 to π but 0.046 to uniform. A4nb0 disables the shipped term; its distance to π falls to 0.037–0.049, its within-layer max/min ratio reaches 5.06–$5.98\times$ against the target's $6.01\times$, and the expert with the smallest target share (0.0128) receives the smallest observed share (0.0168), at $+0.04\%$ validation loss; the $+1.74\%$ difference seen in the training log is the logging artefact noted in §3. The objective therefore shapes utilisation toward the skewed target rather than merely flattening it. Whether matched shares imply class-specific experts is tested in §5. Together, the *where* and *what* interventions move utilisation from near-uniform allocation to a roughly $6\times$-skewed target while offline validation-loss differences stay at $+0.04$–0.05%.

4.1 Downstream evaluation

Table 1 places the fine-tuned arms side by side on the reported downstream measures. Linear probes on frozen features show that the representations carry signal, since every arm clearly beats a 4-mer composition baseline on splice, but no arm separates from A0: the largest gap across the three tasks is $+0.013$ MCC against a bootstrap SD of 0.011. Between-arm coverage differences (0.012 learned, 0.052 k-mer) are smaller than the change within one arm across sweep counts (0.095 and 0.146; Appendix D), so coverage is more sensitive to sampling-chain length than to the utilisation intervention. The pre-registered zero-shot ClinVar endpoint is uninformative under its registered rule (control AUROC at or below 0.55; A0 95% CI 0.497–0.547); the non-coding subset (0.560) was not a registered fallback and is reported descriptively only (Appendix G).

Generation also exposes a model-selection failure. JanusDNA is a denoiser rather than an autoregressive generator, and the Gibbs sampler is not an objective any arm was trained on. Scoring with the same model, A0 scores its own $T{=}0.8$ samples at 0.6342 nats, better than real hg38 (0.8045; uniformly random sequence scores 1.4203), yet those samples lie $10.3\times$ farther from hg38 in 3-mer Jensen–Shannon divergence (0.09383) than $T{=}1.0$ samples from the same sampler (0.009143), which score a worse 1.2587 nats. Two disjoint hg38 samples of the same size give 0.00982, so $T{=}1.0$ sits at this finite-sample floor while $T{=}0.8$ lies about $10\times$ above it. Reconstruction loss alone does not select a realistic genomic generator here.

4.2 Separable but not attributable

A paired bootstrap over the 128 windows (10,000 resamples; nominal, unadjusted 95% intervals) separates 11 of 13 fine-tuned checkpoints from the reference A0 run, with differences as small

	val loss	linear-probe MCC			coverage (16 sweeps)		ClinVar
arm	(nats)	promoter	TF binding	splice	learned	6-mer	AUROC
4-mer composition / real held out	—	0.529	0.580	0.127	0.780	0.881	—
A0 (control)	0.8945	0.569	0.632	0.284	0.396	0.534	0.522
A2	0.8947	0.573	0.637	0.289	—	—	0.524
A4nb	0.8949	0.570	0.637	0.289	—	—	0.521
A4nb0	0.8948	0.574	0.640	0.283	0.390	0.500	0.520
A5nb	0.8949	0.574	0.645	0.286	0.402	0.482	0.521

Table 1: Downstream measures of the fine-tuned arms. The reference row gives the 4-mer composition baseline for the probes and real held-out sequence for coverage. ClinVar is uninformative under the pre-registered rule (control AUROC ≤ 0.55). Probe bootstrap SDs are 0.003–0.012; coverage was computed for the three arms shown (Appendix D).

as $+0.00015$ nats (Appendix A). Two of the 13 are A0 reruns rather than interventions, and the one with a different seed and device split is itself resolved, at $+0.00033$ nats, larger than several intervention effects. With 13 uncorrected intervals, the count is evidence about the set, not about any individual arm. One seed pair is not a variance estimate, but it shows that a resolved difference need not be caused by the intervention. The three-seed replication in §5 concerns layer-7 permutation sensitivity and does not supply further validation-loss seed pairs. **The checkpoints are separable but not attributable with the available seed evidence.** The divergent block-routing arm ($+6.67$ nats) confirms that the harness registers a real effect.

5 Validation loss is sensitive to expert assignment

Our claim concerns relative sensitivity, not that expert load has no effect. On the same 128 windows, all-layer permutation of an A0 run indistinguishable from the reference run (Appendix A) raises validation loss by $7.48 \pm 1.37\%$ over ten permutations (median 7.33%, one outlying permutation at 11.02%), roughly $126\times$ the 0.06% span of the fine-tuned checkpoints. One is an inference-time perturbation and the other a set of fine-tuned objectives, so the ratio is descriptive, not a causal effect estimate. Permutation leaves the utilisation histogram unchanged up to relabelling, so it changes nothing the balancing objective can see: **the quantity the objective regulates and the quantity to which validation loss is most sensitive are different quantities.**

The effect is not an artefact of one indispensable expert or of our fine-tuning. In the four-window decomposition, P costs 7.43%, 97% of fully random routing (7.65%) and $28\times$ the most consequential of 64 single-expert removals (0.27%): removing one expert perturbs only the slots assigned to it, whereas P reroutes 97.4% of selected slots. The released checkpoint, before any fine-tuning, already incurs $7.70 \pm 1.07\%$ on the same 128 windows (median 7.64%) and 7.34% on the four-window protocol, and every JanusDNA checkpoint we permuted on that protocol falls between 6.30% and 7.43%: the sensitivity is not created by our utilisation interventions. Nor are the experts collapsed: the effective rank of their outputs is 12.42–15.78 of a possible 16 (Table 4).

The cost is not mainly a gate-weight mismatch. P also leaves each expert scaled by a gate weight computed for a different expert. We separate the two effects with pre-registered inference-time arms on the four-window protocol (Appendix G): P-gate recomputes gate probabilities for the newly assigned experts, Swap exchanges one expert pair per layer, and Flip keeps the selected experts but swaps their two weights (Table 2).

P-gate retains 89% of the P cost (90–91% on the other JanusDNA checkpoints; 94% on Genos-1.2B). Swap costs 6.7–8.4% per rerouted slot, close to P's 7.6%, so the damage scales with how much assignment changes rather than requiring a globally shuffled network; Flip costs about one fifth of P. Genos-1.2B differs on Flip (86% of its permutation cost), consistent with its renormalised top-2 weights; a mass-displacement normalisation introduced after that result is labelled post hoc (Appendix G). Against the pre-registered reading rules, P-gate exceeds both thresholds on the control (at least 5% and at least two thirds of P), Swap's per-slot cost is within a factor of two of P's on every JanusDNA checkpoint, and Flip is 15–22% of P (Appendix G).

A0 intervention	Δ val loss	slots rerouted	loss per rerouted slot
P	$+7.43\%$	97.4%	7.6%
P-gate	$+\mathbf{6.64}\%$	97.5%	6.8%
Swap one expert pair	$+0.82$–1.31%	12–16%	6.7–8.4%
Flip (weights only)	$+1.50\%$	0	—

Table 2: Decomposing the all-layer permutation on A0 (four-window protocol).

checkpoint	per-layer term	L1	L3	L5	L7	L7, three seeds
released	no	0.26%	3.13%	4.21%	2.18%	—
A0z (no balancing)	no	0.29%	3.17%	4.21%	2.23%	—
A0 (control)	no	0.30%	3.11%	4.10%	2.35%	$2.35/2.21/2.21\%$
A2	no	0.26%	3.02%	3.92%	2.19%	—
A4nb (uniform outcome)	yes	0.24%	2.96%	3.80%	$\mathbf{1.38}\%$	—
A4nb0 (prior target)	yes	0.23%	3.03%	3.95%	$\mathbf{1.37}\%$	$1.37/1.37/1.37\%$
A5nb	yes	0.22%	3.05%	3.92%	$\mathbf{1.35}\%$	—

Table 3: Single-layer permutation sensitivity (four-window protocol). Layer 7 separates arms with and without a per-layer term far more than layer 1; seeds rounded to two decimals.

	class enrichment $\lvert \log_2 \rvert$				effective rank (of 16)			
checkpoint	L1	L3	L5	L7	L1	L3	L5	L7
released	1.94	1.41	1.66	4.31	15.78	14.64	14.48	13.81
A0z (no balancing)	1.83	1.51	1.94	4.72	—	—	—	—
A0 (control)	2.19	1.53	1.77	4.52	15.72	14.61	14.14	14.30
A4nb (uniform outcome)	1.20	0.90	0.79	2.74	—	—	—	—
A4nb0 (prior target)	0.92	0.92	**2.26**	3.11	15.42	14.36	13.57	**12.42**

Table 4: Class enrichment against a within-window matched null (99th percentile 0.16–0.29) and effective rank of the stacked expert outputs on 2,048 shared token representations. Rank was measured for the three checkpoints shown.

Arms with a per-layer term show lower layer-7 assignment dependence. Single-layer permutation depends strongly on depth (Table 3): layer 5 is the most sensitive, and the all-layer penalty is about 75% of the sum of the single-layer penalties, so layer contributions overlap. Layer 7 separates the checkpoints by objective far more strongly than any other layer (layer 1 shows a much smaller non-overlapping difference). The four without a per-layer term average 2.24% and the three with it 1.37%, about 39% lower, with no overlap. The cumulative ladder makes one contrast controlled: A4nb differs from A2 only by the per-layer prior-targeted term, with the reverse-complement, smoothness and shipped coefficients unchanged, and its layer-7 sensitivity falls from 2.19% to 1.38%. Switching off the shipped term adds nothing further (A4nb0 1.37%, A5nb 1.35%), and the reverse-complement and smoothness terms do not produce the reduction, since A2 carries both and stays within the control range; A0z, which removes balancing entirely during fine-tuning, also stays there. The separation also holds across three seeds of A0 and A4nb0. Because no per-layer arm was trained with a uniform target, this association does not isolate the effect of layer-wise aggregation from that of the prior target. This is the clearest structural separation among the trained arms, obtained where validation loss, probes and generation show no attributable difference: the objective is not inert. We read the lower penalty as reduced assignment dependence, not demonstrated performance degradation (§8).

Class specificity and output diversity move with it. Experts are class-specific before fine-tuning: in the released checkpoint the strongest enrichment reaches $\lvert \log_2 \rvert = 4.31$ at layer 7, against a matched-null 99th percentile of 0.16–0.29 (Table 4). Both prior-targeted arms reduce it at layer 7, while A0z, trained without the shipped term, is the most class-specific. Relative to A0, A4nb0 lowers enrichment at layers 1, 3 and 7 but raises it at layer 5, and lowers effective rank at layers 5 and 7, where mean pairwise cosine also rises (layer 7: 0.07 to 0.10). Matching a composition-derived share therefore does not make an expert specific to the class the share was derived from. The three measures are computed from different objects; the concurrent drops at layer 7 are consistent with

less differentiated experts, but neither establish a causal link among the measures nor demonstrate predictive harm. Enrichment and rank use one checkpoint per arm and were not measured on the seed replicates.

6 Routing structure persists across genomic MoEs

We repeat the inference-time interventions on PlantBiMoE and Genos-1.2B, which differ from JanusDNA in expert count, routing sparsity, backbone and training objective, and, for Genos-1.2B, in being $34\times$ larger.

	JanusDNA (A0) 35.6 M, E=16, k=2 Mamba, denoiser	PlantBiMoE 140.8 M, E=4, k=1 Mamba, MLM	Genos-1.2B 1.2 B, E=8, k=2 Transformer, AR
permutation	$+7.43\%$	$+122.8\%$	$+134.5\%$
random routing	$+7.65\%$	$+39.8\%$	$+134.9\%$
worst single-expert ablation	$+0.27\%$	$+83.4\%$	$+73.0\%$

Table 5: Inference-time interventions in three genomic MoEs, each on its own protocol; compare within columns only.

Within each model, permutation increases loss more than the most consequential single-expert ablation (Table 5). Magnitudes are not comparable across columns: the models use different corpora and protocols (JanusDNA: the four-window decomposition; PlantBiMoE: 32 masked 512 bp TAIR10 windows with five permutation draws, five random-routing draws and four single-expert ablations, not verified to be disjoint from pretraining; Genos-1.2B: $16 \times 8{,}192$ tokens; Appendix I). Three JanusDNA observations do not transfer. First, single experts are critical elsewhere (83.4% and 73.0%), so the shared result is the ordering, not that experts are individually unimportant. Second, permutation matches random routing in the two top-2 models but costs $3.1\times$ random routing in top-1 PlantBiMoE, where a random draw still reaches a token's original expert with probability $1/4$ while a global permutation moves all tokens of a non-fixed expert together; expert count and sparsity are not isolated here. Third, the decomposition transfers only in part: per-slot swap costs are not comparable across models (Appendix G). Finally, utilisation cannot be moved post hoc while holding assignment fixed: a router bias that shifts the Genos-1.2B histogram only $1.41\times$ already reassigns 16.9% of tokens and raises loss by 47.4%, with larger biases reassigning more tokens and costing more (Appendix I), so the retraining comparison of §4 has no inference-time analogue.

7 Exploratory structure-aware objectives

Reverse complementation combines sequence reversal and base complementation, so we apply each separately to the same forward router. Against the matched null, reversal retains substantial agreement at all four MoE layers ($\kappa = 0.23$–0.54 at 128 and 512 bp), consistent with the architecture's reversal path, whereas complementation is near null (-0.06 to $+0.06$; Appendix A). The gap therefore lies along complementation, and a reversal-only constraint would target an axis that is already comparatively consistent. Pooled reverse-complement agreement at 128 bp is $\kappa = 0.2201$ against the uniform null, in the pre-registered *partial* band, and 0.1065 against the matched null, reported as a sensitivity analysis because the thresholds were registered against the uniform null. A block-level Jensen–Shannon penalty between routing on a sequence and on its reverse complement (A1; Appendix C), computed on block-pooled routing distributions while the router itself still assigns individual tokens, raises complement-only agreement by $+0.327$ against $+0.012$ for reversal and -0.008 for forward–backward router agreement, at $1.53\times$ the training-step time, in a single-checkpoint comparison without established downstream benefit. The pre-registered rule did not select its coefficient: every value passed, and the largest-value tiebreak chose $\lambda = 0.20$, which averages 1.9 inactive experts at layer 7, whereas $\lambda = 0.05$ has none. An attribution-based assignment objective (A6) raised a layer-7 best-in-selection proxy from 7.2% to 27.2%, but only to 10.2%, below the 12.5% chance level, once expert-output magnitude was removed: it optimised a magnitude-sensitive proxy rather than a demonstrated routing criterion (Appendix B). Block-level routing did not train

stably in five attempts; a centred pooling rule started near the control loss but diverged with every expert still active, so expert death is not necessary for the failure (Appendix E).

8 DISCUSSION

If moving utilisation had an attributable validation-loss cost, it would define a trade-off. Instead, utilisation moved across a $6\times$ within-layer range at no attributable cost, which raises a more basic question: what does the model's behaviour depend on? A relabelling the objective cannot see cost 7.48%. Token–expert correspondence is a distinct, loss-sensitive property that load balancing does not directly regulate. The ordering replicates in two further genomic MoEs; the layer-7 reduction in arms with a per-layer prior-targeted term has been measured only in JanusDNA.

What to monitor alongside utilisation. Per-layer utilisation and dead-expert counts remain useful training logs, and a pooled statistic can hide the failure. At selected checkpoints we propose (i) permutation sensitivity, (ii) class enrichment against a within-window matched null, and (iii) the effective rank of expert outputs on shared inputs. They require separate evaluation and are neither validated downstream surrogates nor replacement objectives. A change in one of them is a prompt to investigate a checkpoint, not a score of biological competence: a fall in permutation sensitivity, for instance, may mean that experts became less differentiated or that redundancy made routing mistakes less costly, and class enrichment and output geometry are complementary evidence that cannot resolve this without an independent task endpoint. The case for measuring them is the one our ladder illustrates: in A4nb the per-layer term reduced layer-wise Gini 3.6–$8.8\times$, which a practitioner monitoring only the histogram would have recorded as an improvement, yet the same arm's layer-7 permutation sensitivity fell from 2.35% to 1.38% and its class enrichment is lower than A0's at every layer (Tables 3 and 4). Sampled pair swaps are cheaper but need per-model calibration against a full permutation. No general warning threshold is established; checkpoints should be compared on the same inputs and protocol (Appendix H).

When to adjust balancing. Nothing here argues for removing the objective: PlantBiMoE and Genos-1.2B, whose single experts are critical (§6), may be in exactly the regime it was designed for, although our data do not establish whether balancing caused or prevented that concentration. Once no expert is dead and single-expert ablations are cheap, as in JanusDNA, continuing to regulate the histogram moves a quantity to which validation loss is comparatively insensitive. Annealing balancing at that point is a hypothesis we have not tested; A0z, fine-tuned from a checkpoint pretrained with the term, is consistent with it but does not test it, since it shows only that this pretrained router stays active without the auxiliary term during fine-tuning. The open question is when and how to adjust balancing, not whether to discard it (Appendix H).

Limitations. *Seeds and attribution.* The fine-tuning grid has one primary seed: the validation-loss retraining control is one seed pair, the layer-7 permutation result has three seeds for A0 and A4nb0, and enrichment and rank were not measured on replicates. Bootstrap comparisons are nominal and uncorrected. *Scope.* All arms fine-tune one 35.6 M checkpoint on human sequence for 10,000 steps; the effect of these objectives during pretraining is untested. The other two models support the permutation-over-ablation ordering, not JanusDNA's magnitudes. *Measurement.* JanusDNA losses are unmasked reconstruction-style scores; some routing and generation statistics use small samples (four or eight windows; eight 4,096 bp sequences); PlantBiMoE windows were not verified to be disjoint from pretraining; the zero-shot variant endpoint was uninformative, and the RC selection rule did not discriminate. None of the structural measurements alone establishes predictive benefit or harm.

REPRODUCIBILITY STATEMENT

JanusDNA validation-loss values for the fine-tuned arms are recomputed offline from stored checkpoints, rather than read from training logs, using the same 128 held-out windows in the same order, in one process on one GPU (§3). The separately measured all-layer permutation mean is also reported on those same 128 windows ($+7.48\%$; A0 d8a1 baseline 0.89449 nats), while the P-gate, random-routing, and ablation decomposition retains its original four-window protocol ($+7.43\%$

for P). Cross-model interventions use their own protocols: PlantBiMoE uses 32 sampled TAIR10 chromosome-1 windows of 512 bp with a fixed 15% mask, and Genos-1.2B uses $16 \times 8{,}192$ tokens (§6). The PlantBiMoE windows have no verified exclusion from pretraining data and are not reported as held-out performance. §3 states the JanusDNA checkpoint (with its DOI and verified md5), the architecture, and the training configuration shared by its arms; Appendix A gives every table the main text summarises, including the `(devices, accumulation)` configurations whose effect on logged validation loss we describe. Section 5's permutation, single-expert ablation, and random-routing interventions act at inference on fixed checkpoints and require no training to reproduce; the main comparison uses A0, with the released checkpoint reported alongside as an additional fixed-checkpoint control (not an independent retraining replicate). The pre-registered thresholds and selection rules are stated where each is applied, together with the outcomes of the two that did not behave as registered. The supplementary material lists every claim a control overturned, and what overturned it. Code, per-arm diagnostics, the evaluation harness and the pre-registration document are released.

USE OF GENERATIVE AI

Generative AI tools were used for language editing, drafting parts of the manuscript, brainstorming, literature search, and assistance with software implementation and experimental analysis. All experiments and reported results were independently executed and verified by the authors, who take responsibility for the final manuscript and its conclusions.

REFERENCES

Žiga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle R. Taylor, Tom Ward, et al. Advancing regulatory variant effect prediction with AlphaGenome. *Nature*, 649(8099):1206–1218, 2026. doi: 10.1038/s41586-025-10014-0.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652(8112): 1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Hugo Dalla-Torre et al. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 2025.

Qihao Duan, Bingding Huang, Zhenqiao Song, Irina Lehmann, Lei Gu, Roland Eils, and Benjamin Wild. JanusDNA: A powerful bi-directional hybrid DNA foundation model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.17257.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V. Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3929–3938, 2019.

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

Adi Lin, Bin Xie, Cheng Ye, Cheng Wang, Duoyuan Chen, Ercheng Wang, Fanfeng Lu, Guirong Xue, Haiqiang Zhang, Jiajie Zhan, et al. Genos: a human-centric genomic foundation model. *GigaScience*, 14:giaf132, 2025a. doi: 10.1093/gigascience/giaf132.

Kepeng Lin, Qizhe Zhang, Rui Wang, Xuehai Hu, and Wei Xu. PlantBiMoE: A bidirectional foundation model with SparseMoE for plant genomes. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2025b. arXiv:2512.07113.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning (ICML)*, pp. 7176–7185, 2020.

Eric Nguyen et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2699–2712, 2020.

Yair Schiff et al. Caduceus: Bi-directional equivariant long-range DNA sequence modeling. In *International Conference on Machine Learning (ICML)*, 2024.

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

Alexandros Tzanakakis, Ioannis Mouratidis, and Ilias Georgakopoulos-Soares. Fundamental limitations of genomic language models for realistic sequence generation. *bioRxiv*, 2026. doi: 10.64898/2026.01.17.700093.

Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a markov random field language model. In *Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, 2019.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

Zhao Yang, Jiwei Zhu, and Bing Su. SPACE: Your genomic profile predictor is a powerful DNA foundation model. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2506.01833.

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

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. In *International Conference on Learning Representations (ICLR)*, 2022.

A EXTENDED TABLES AND FIGURES

Validation loss, main arm grid. Recomputed from the step-10,000 checkpoints over the same 128 held-out windows in one process. A3c is included as a positive control to check that the evaluation harness can detect a clearly different checkpoint. The non-divergent band covers the fourteen checkpoint evaluations of Table 6: three A0 runs, A1 at three coefficients, A4nb0 at two seeds, and one each of A1w, A2, A4nb, A5nb, A6 and A6h. A3c is a divergent positive control and is excluded from the 0.06% band. A0z is reported under a separate comparison protocol, as noted below. The paired-bootstrap comparisons of §4.2 are listed individually in Table 6.

checkpoint	run ID	val loss	Δ vs A0	95% CI	excludes 0
A0 s2222 d8a1†	`A0_lr8e-4_h100nockpt`	0.894494	+2.5	[−5.7, +10.5]	–
A0 s2223 d2a4†	`A0_lr8e-4_s2223_h100nockpt_g2`	0.894799	+33.0	[+12.5, +53.2]	✓
A1 λ=0.0117	`A1_lam0.0117_lr8e-4`	0.894492	+2.3	[−5.6, +10.0]	–
A1 λ=0.05	`A1_lam0.05_lr8e-4`	0.894622	+15.3	[+7.2, +23.3]	✓
A1 λ=0.20	`A1_lam0.20_lr8e-4`	0.894956	+48.7	[+38.0, +59.3]	✓
A1w	`A1w_lam0.05_lr8e-4_h100nockpt`	0.894626	+15.6	[+7.3, +24.3]	✓
A2	`A2_lam0.05_lr8e-4`	0.894660	+19.0	[+10.4, +27.6]	✓
A4nb d4a2	`A4nb_lam0.05_lr8e-4_h100nockpt`	0.894881	+41.2	[+32.5, +49.7]	✓
A4nb0 s2222	`A4nb0_lam0.05_lr8e-4_h100nockpt_g2`	0.894821	+35.2	[+24.0, +45.9]	✓
A4nb0 s2223	`A4nb0_lam0.05_lr8e-4_s2223_h100nockpt_g2`	0.895001	+53.2	[+37.0, +67.2]	✓
A5nb d2a4	`A5nb_lam0.05_lr8e-4_h100nockpt_g2`	0.894853	+38.3	[+29.4, +47.5]	✓
A6	`A6_lr8e-4_a6`	0.894933	+46.4	[+30.2, +65.5]	✓
A6h	`A6h_lr8e-4_a6h`	0.894723	+25.4	[+9.9, +43.0]	✓

Table 6: Paired bootstrap of fixed-window validation loss against the reference A0 run (`A0_lr8e-4`, s2222 d2a2, 0.894469 nats); Δ and CI in 10^{-5} nats; 10,000 resamples (seed 0) shared across comparisons. †A0 reruns (different seed or device split), not interventions; the seed rerun already excludes zero, which is the direct basis for "separable but not attributable". Intervals are nominal and unadjusted for multiple comparisons; their exclusion of zero should not be read as family-wise evidence for individual intervention effects, so the count is evidence about the set, not about any individual arm. The divergent positive control A3c ($\Delta = +667{,}060$) is excluded.

arm	val LM	vs control
A0 (control, three configurations)	0.8945 / 0.8945 / 0.8948	—
A1 λ=0.0117 / 0.05 / 0.20	0.8945 / 0.8946 / 0.8950	+0.00 to +0.05%
A1w (depth-weighted)	0.8946	+0.02%
A2 (+ smoothness)	0.8947	+0.02%
A4nb / A4nb0 / A5nb	0.8949 / 0.8948 / 0.8949	+0.04–0.05%
A4nb0 s2223 / A6 / A6h	0.8950 / 0.8949 / 0.8947	+0.03 to +0.06%
A3c (divergent, positive control)	7.5651	+746%
A0z was measured on permutation sensitivity and class specificity rather than on this fixed set.		

Matched 128-window permutation measurement. The permutation experiment loads the A0 d8a1 run (baseline 0.89449 nats), which is not distinguishable from the reference A0 run on the same windows ($\Delta = +2.5 \times 10^{-5}$, CI $[-5.7, +10.5] \times 10^{-5}$; Table 6); the mean all-layer permutation penalty is $+7.48 \pm 1.37\%$ over 10 permutations (median $+7.33\%$; `results/perm128.json`). It shares the evaluation inputs with the fine-tuned checkpoint grid. The original $+7.43 \pm 1.63\%$ estimate comes from `probe_redundancy.py` on four 131,072 bp windows with five permutations and remains the reference for the intervention decomposition.

The validation-configuration artefact. Logged validation loss at step 10,000 by `(devices, accumulation)`, all at global batch 8. The three runs at $(2, 4)$ are two different arms at two different seeds; seed variation within that configuration is 0.12%.

configuration	$(2, 2)$	$(4, 2)$	$(8, 1)$
logged val LM	0.9006	0.9008	0.9016
configuration		$(2, 4)$	
logged val LM	0.9172	0.9173	0.9184

Per-layer utilisation against the two targets.

arm	quantity	L1	L3	L5	L7	spread
A4nb	L_1 to π	0.200	0.202	0.212	0.214	—
	L_1 to uniform	**0.046**	**0.041**	**0.044**	**0.056**	—
	max/min	1.25	1.28	1.31	1.26	1.26×
	Gini	0.032	0.029	0.031	0.038	—
A4nb0	L_1 to π	**0.038**	**0.037**	**0.048**	**0.049**	—
	L_1 to uniform	0.227	0.215	0.195	0.215	—
	max/min	5.98	5.10	5.39	5.06	target 6.01×
	Gini	0.153	0.143	0.134	0.151	Gini(π) = 0.142
A0	Gini	0.221	0.104	0.194	0.334	—

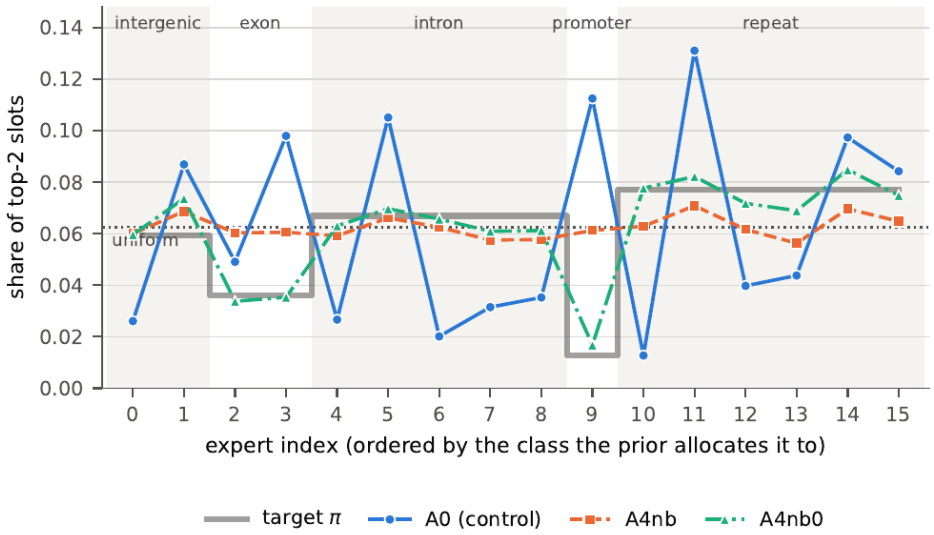

Figure 3: Per-expert utilisation at layer 7 against the two candidate targets, experts ordered by the class the prior allocates them to. The control is strongly uneven but not aligned with π (10.3× spread in a different arrangement); A4nb, with the shipped term still running, collapses to 1.26×; A4nb0 tracks π across every class at 5.06×, including the single expert the prior allocates to promoter, which receives the smallest share of the sixteen.

Permutation cost by layer. Δ validation loss when a single layer's expert assignment is permuted. The four checkpoints without per-layer balancing and the three with it do not overlap at layer 7.

checkpoint	per-layer balancing	L1	L3	L5	L7
released	no	0.26%	3.13%	**4.21%**	2.18%
A0z (*no* balancing)	no	0.29%	3.17%	**4.21%**	2.23%
A0 (control)	no	0.30%	3.11%	**4.10%**	2.35%
A2	no	0.26%	3.02%	**3.92%**	2.19%
A4nb	yes	0.24%	2.96%	**3.80%**	**1.38%**
A4nb0	yes	0.23%	3.03%	**3.95%**	**1.37%**
A5nb	yes	0.22%	3.05%	**3.92%**	**1.35%**

Class specificity and expert diversity. Strongest class enrichment $|\log_2|$ against a within-window assignment-permuting null (99th percentile 0.16–0.29), and effective rank / mean pairwise cosine of the stacked expert outputs on 2,048 held-out token representations. Effective rank is the singular-value participation ratio $r_{\text{eff}} = (\sum_i s_i)^2 / \sum_i s_i^2$ of these output matrices, not of expert-weight matrices.

enrichment $\lvert \log_2 \rvert$	L1	L3	L5	L7
released	1.94	1.41	1.66	**4.31**
A0z (no balancing)	1.83	1.51	1.94	**4.72**
A0 (control)	2.19	1.53	1.77	**4.52**
A4nb (uniform outcome)	1.20	0.90	0.79	2.74
A4nb0 (prior target)	0.92	0.92	2.26	3.11
effective rank / cos	L1	L3	L5	L7
released	15.78 / 0.06	14.64 / 0.22	14.48 / 0.21	13.81 / 0.06
A0 (control)	15.72 / 0.07	14.61 / 0.22	14.14 / 0.25	14.30 / 0.07
A4nb0 (prior target)	15.42 / 0.07	14.36 / 0.24	**13.57** / 0.29	**12.42** / 0.10

Reverse-complement axis decomposition. κ against the within-window matched null under each transformation applied alone to the same forward router, at two block sizes.

	complement only		reversal only		reverse complement	
layer	b128	b512	b128	b512	b128	b512
1	−0.028	−0.064	0.426	0.539	−0.002	−0.027
3	+0.009	+0.041	0.372	0.515	+0.065	+0.092
5	+0.023	+0.037	0.231	0.306	+0.124	+0.126
7	+0.024	+0.057	0.343	0.332	+0.239	+0.321

Block-size sweep. Contrast shrink under pooling, against the $\sqrt{b}$ that independent samples would give, and the mode-pair share of the two pooling rules at layer 7.

block	16	32	64	128	256
contrast shrink	1.88×	2.13×	2.39×	2.72×	3.09×
$\sqrt{b}$	4.00×	5.66×	8.00×	11.31×	16.00×
implied length	4.5	7.1	11.2	**17.4**	26.9
mean-prob, mode-pair	15.5%	18.8%	22.8%	27.0%	33.5%
centred, mode-pair	9.1%	8.3%	8.5%	**8.3%**	8.4%

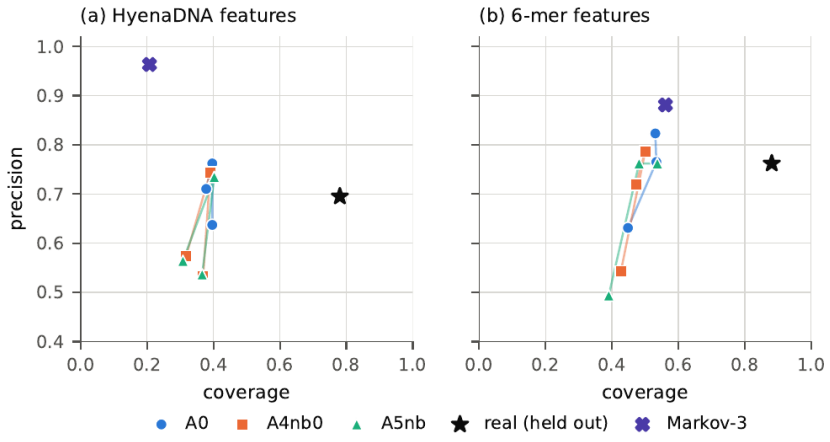

Figure 4: Generated sequence in the coverage–precision plane, 512 samples per arm, three points per arm for 8, 16 and 32 Gibbs sweeps. **No arm separates from the control here** — they overlap in both feature spaces, and the largest gap between them is smaller than the spread each covers on its own as its chain runs longer. All sit far from real held-out sequence (★) with recall collapsed. The Markov-3 control (×) separates the two spaces: the learned encoder places it well below every arm, while in 6-mer space it *exceeds* them all on coverage.

Router bias sweep, Genos-1.2B. A per-expert bias $b_e \sim \mathcal{N}(0, \sigma^2)$ added to the router logits at inference, 8,192 tokens. It cannot isolate the utilisation axis: the selected set moves with the histogram, and the loss follows the former.

σ	histogram spread	Δ loss	tokens reassigned	both experts changed
0.25	$1.41\times$	$+47.4\%$	16.9%	0.1%
0.50	$1.99\times$	$+65.8\%$	30.3%	0.4%
1.00	$4.16\times$	$+71.8\%$	53.5%	2.7%
2.00	$62.2\times$	$+100.5\%$	82.0%	21.4%

For reference, the full expert-identity permutation changes 82.2% of selected expert slots and costs $+134.5\%$: 53.5% reassignment for 53% of that cost is close to proportional.

Gibbs sampler mixing. 3-mer JS divergence to hg38 from two initialisations, and the distance between the two chains, at 8, 16 and 32 sweeps. The chains converge on each other while both move away from the data.

sweeps	from uniform	from real	between the chains
8	0.00503	0.00243	0.00836
16	0.00484	0.00306	0.00311
32	0.00685	**0.00830**	**0.00147**

GC agrees: 0.407 and 0.500 at the two starts, 0.342 and 0.355 at the end.

Generation and probes. Coverage at 16 sweeps in both feature spaces, and linear-probe MCC on three tasks.

	HyenaDNA space		6-mer space	
set	coverage	promoter	coverage	promoter
real (held out)	**0.780**	**0.783**	**0.881**	**0.870**
A0 (control)	0.396	0.304	0.534	0.391
A4nb0	0.390	0.348	0.500	0.304
A5nb	0.402	0.261	0.482	0.348
3rd-order Markov	0.207	0.130	0.561	0.304

MCC	promoter core	TF binding	splice (3-way)
4-mer composition	0.529 ± 0.006	0.580 ± 0.012	0.127 ± 0.007
A0 (control)	0.569 ± 0.004	0.632 ± 0.012	0.284 ± 0.007
A2	0.573 ± 0.005	0.637 ± 0.010	0.289 ± 0.005
A4nb	0.570 ± 0.003	0.637 ± 0.011	0.289 ± 0.009
A4nb0	0.574 ± 0.005	0.640 ± 0.010	0.283 ± 0.007
A5nb	0.574 ± 0.006	0.645 ± 0.011	0.286 ± 0.006

B THE ASSIGNMENT-REGRET OBJECTIVE

DIAGNOSIS, AND WHY THE ATTRIBUTION FAILS AT LAYER 7

If assignment is what the model depends on, the objective that follows should regulate assignment rather than the histogram. Whether one is worth building is an empirical question we can settle without training: if the router already selects the best available expert for almost every token, an assignment-aware term can only do harm. For token x with block output y and expert outputs $f_i(x)$, let $a_i = \langle \partial L/\partial y,\ f_i(x) \rangle$ be a first-order attribution score, lower being better. The all-expert diagnostic reported below compares the selected top-2 against the best score among all 16; its regret is $r = \left[\min_{i \in \text{sel}} a_i - \min_{j \in \{1,\dots,16\}} a_j \right]_+$. The training objective, specified below, instead samples $m{=}4$ unselected experts on $\rho{=}5\%$ of tokens to limit computation. On the released checkpoint over held-out tokens:

	L1	L3	L5	L7	null
$P(\text{regret} > 0)$, raw	87.7%	81.4%	82.5%	**93.7%**	87.5%
$P(\text{regret} > 0)$, direction only	88.2%	79.4%	75.3%	**90.9%**	87.5%
$r \,/\, \mathrm{sd}(a)$ on the same token	0.72	0.38	0.62	**1.03**	—

The null matters more here than anywhere else in this paper. Because regret compares the minimum over the two selected experts against the minimum over all sixteen, an uninformative router leaves the global best unselected with probability $1 - 2/16 = 87.5\%$. Read against an implicit null of zero, 79–94% would suggest headroom (supplementary material); against the right null, layer 1 is at chance, layers 3 and 5 are modestly better than chance, and **layer 7 is worse than chance**. The complementary statistic, how often the router's top choice is the best available expert, is $1 - P(\text{regret} > 0)$ exactly and is therefore not reported separately.

Layer 7 shows the weakest attribution-based ranking. The loss gradient arriving at the block shrinks with depth, from 7.3×10^{-3} at layer 1 to 3.3×10^{-4} at layer 7, a factor of 22, coinciding with the poorer ranking. Gradient magnitude alone cannot explain it: scaling g by a positive constant scales every attribution $\langle g, f_i \rangle$ equally and leaves ranks and standardised regret unchanged. One possible explanation, which we have not tested, is that where the first-order term is small, higher-order terms make up a larger share of the true counterfactual loss change, so the first-order ranking predicts it less well. The displayed normalised regret, $r/\mathrm{sd}(a)$, is highest at layer 7 (1.03), but this contrast does not by itself establish estimator bias or exclude noise, and spread in expert-output norms does not explain the anomaly, since two layers with identical norm spread differ by six points. The mechanism remains unresolved.

The same shape reproduces at half the coefficient (below), so this is a property of the measurement and not of one run. We therefore report the layers where the attribution behaves — 3 and 5, modestly better than chance — and do not claim headroom at layer 7. The objective this motivated is specified at the end of this appendix and its arm reported in the next subsection; the diagnosis does not license the cure, and the cure turns out to illustrate the paper's own thesis rather than escape it.

ONLY THE LANGUAGE-MODELLING COST TRACKS THE COEFFICIENT

We built the objective specified below (arm A6), and then built it again at half the coefficient (A6h, $\lambda = 0.01325$, otherwise identical). The natural defence against any result here is that the coefficient was simply mis-set, so measuring two of them is the point. Validation-loss costs in this table use A0 d8a1 as the comparator; Table 6 uses the reference A0 run.

quantity	A6 (λ)	A6h ($\lambda/2$)	retained
validation LM cost vs A0	$+44 \times 10^{-5}$	$+23 \times 10^{-5}$	**52%**
95% CI, paired bootstrap	—	$[+7, +41]$	—
permutation-sensitivity reduction, all four layers	1.72	1.53	89%
permutation-sensitivity reduction, layer 7	1.48	1.33	90%
class-enrichment loss, layer 7	0.83	0.93	112%
class-enrichment loss, layer 5	0.84	0.83	99%
direction-only regret gain, layer 5	+5.0	+4.1	82%
direction-only regret gain, layer 3	+1.6	+2.0	125%

Halving the coefficient halves the language-modelling cost and leaves everything else where it was. The LM term retains 52%, close to proportional; the benefit and the structural reduction retain 78–125%. Two points cannot show that the effect has saturated — both may lie on a plateau — but they do show it is not proportional to the coefficient over a two-fold range, and that the two tested coefficients did not decouple the apparent benefit from the accompanying structural reduction. Intermediate coefficients remain untested, so we do not exclude a different trade-off between them.

The asymmetry is the result: **the one quantity that responds to the coefficient is the one a practitioner would be watching.** Tuning λ down buys an arm that looks half as expensive and does the same damage — the paper's thesis one level up, with the monitored and the affected quantity again being different quantities. Layer 1 is the internal control: its per-layer weight is small enough to leave the constraint effectively off, and it shows neither a reduction (0.30%, the control's value) nor gain at either coefficient. The constraint is a matter of whether it engages, not of how hard.

And the gain it appears to buy at layer 7 is an artefact of the same attribution. A6 raises layer-7 best-in-selection from 7.2% to 27.2%, which reads as the intervention working exactly where the diagnosis said it should; with expert-output magnitude removed it is 10.2%, below the 12.5% chance level, and A6h reads 20.9% raw against 9.1% direction-only. The apparent improvement is movement toward large-magnitude experts, reproduced at two coefficients: the objective optimised the proxy rather than what the proxy stood in for. The class-enrichment inversion at layer 7 (0.83 against 0.93) is three percent on one run per coefficient and is not resolved, and expert diversity was not re-measured at $\lambda/2$.

The assignment objective used by A6 and A6h is specified below. Its gradient share was calibrated on the released checkpoint; the trained-arm results are reported above.

For each MoE block, on a random $\rho{=}5\%$ of tokens, draw a probe of $m{=}4$ unselected experts, compute the counterfactual attributions $a_i = \langle \partial L/\partial y,\, f_i(x) \rangle$ for the selected and probed experts, and form

$$w_j = \frac{\big[\min_{i\in\mathrm{sel}} a_i - a_j\big]_+}{\mathrm{sd}\big(\{a_i\}_{i\in\mathrm{sel}}\big)}, \qquad \mathcal{L}_{\mathrm{assign}} = -\,\mathbb{E}\big[\,\mathrm{sg}(w_j)\,\log \mathrm{softmax}(z/\tau)_j\,\big],$$

where z are the router logits, τ a temperature and sg the stop-gradient. It is a weighted cross-entropy pulling router mass toward experts a first-order estimate says would have done better, weighted by how much better and normalised by how much the choice matters on that token.

Three properties are deliberate. The weight is detached, so the term shapes the router and never back-propagates into the experts through the attribution. Standardisation is load-bearing rather than cosmetic, for the reason §5 gives: without it the term reads 0.000 against the language model. And the probe is sampled rather than exhaustive — at $m{=}4$ and $\rho{=}5\%$ the measured overhead is 4.0% additional expert evaluations per token, against 700% for scoring all sixteen. Attributions for the *selected* experts are free: we verified numerically that this comparison is the language model's own gradient arriving at the router logits through the softmax Jacobian, so only the probed experts cost anything.

The coefficient is set by the rule used for every other constraint here, gradient share at the router rather than loss share. The term's gradient is $7.55\times$ the language model's, so a 20% target share gives $\lambda_{\mathrm{assign}} = 0.0265$. Per-layer weights $(0.30,\ 3.11,\ 4.10,\ 2.35)$ are the measured single-layer permutation costs of Appendix A, which spends the constraint in proportion to what each layer's assignment is worth.

C THE REVERSE-COMPLEMENT CONSTRAINT

Pre-registered band. At block size 128 bp, pooled reverse-complement agreement is $\kappa = 0.2201$ against the uniform null, in the pre-registered *partial* band (0.1143–0.4000), and 0.1065 against the matched null, just below that band's lower bound. Because the thresholds were registered against the uniform null, the matched-null value is a sensitivity analysis, not a second categorical verdict.

Figure 5 shows the backbone and the point at which this term and the balancing term of §4 act. The backbone is unmodified: every constraint attaches to the router through a forward hook, and none to the experts or the mixture. Note in particular that the loss pools while the router does not — pooling the router's *input* is a separate intervention, arm A3, and is a negative result (Appendix E).

Let p_b^f and p_b^r be routing distributions pooled over block b on the forward pass and on $\mathrm{RC}(x)$; since reverse complementation reverses block order, $\mathcal{L}_{rc} = \frac{1}{B}\sum_b \mathrm{JSD}(p_b^f \,\|\, p_{B-1-b}^r)$. JSD is symmetric and bounded by $\ln 2$, which keeps the term from dominating early training. **The loss pools; the router does not.** Token-level routing follows base identity, so a per-token version would require sending A and its complement T to the same expert, discarding the signal the router uses; over a window the requirement is well posed. The term costs a second forward pass, $1.53\times$ per step. Weights are set by gradient share at the router rather than loss share — at block 128 the term is 0.040% of the loss value and 4.3% of the LM gradient, because the LM objective reaches the router only through the expert mixture while the constraint acts on the logits directly, so calibrating from loss share would have overshot by $\approx 50\times$.

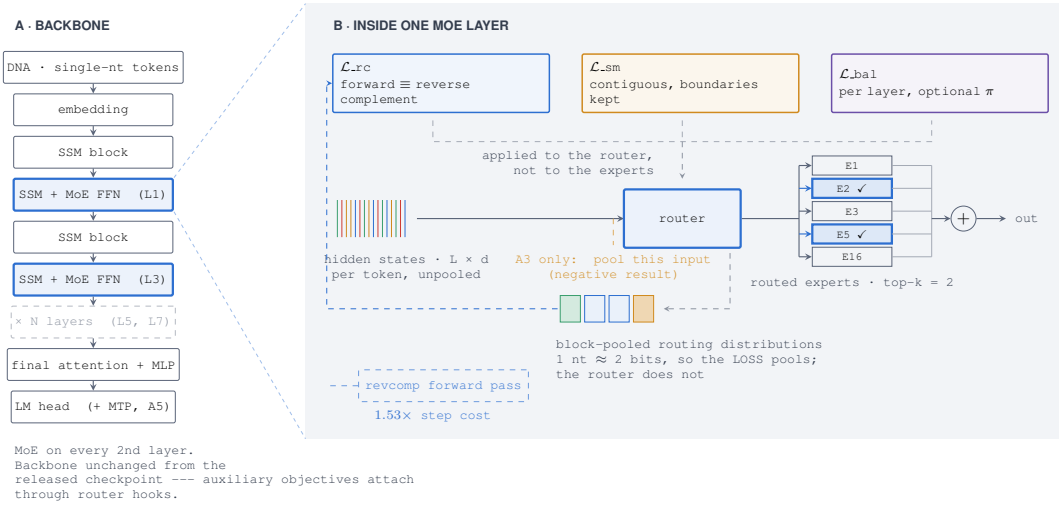

Figure 5: **Where the constraints attach. (a)** The backbone is the released checkpoint, unmodified: MoE feed-forward layers on every second layer and a final attention block without experts. **(b)** Every constraint attaches to the router; expert parameters and the mixture are untouched. The router reads *unpooled* per-token hidden states — token-level routing follows base identity, so a per-token strand constraint would ask it to discard the signal it uses — while the RC and smoothness losses are computed on block-pooled routing distributions and balancing is computed per layer. Pooling the router's *input* is a separate intervention, arm A3, and is a negative result (Appendix E). $\mathcal{L}_{rc}$ needs a second forward pass on the reverse complement, the main added compute cost; $\mathcal{L}_{sm}$ penalises jitter between adjacent blocks; $\mathcal{L}_{bal}$ replaces the pooled uniform objective with a per-layer one targeted at measured composition (§4). Tokens and pooled blocks are tinted by base.

§7 identifies complementation as the initially weaker axis before the constraint existed, and the constraint moves that axis and no other: Δ(A1$-$A0) is $+0.327$ on complement, $+0.012$ on reversal and -0.008 on reading direction. It primarily improves the initially weaker complementation axis rather than uniformly increasing all agreement measures.

We had expected the opposite (this was not a separate pre-registered prediction), since a collapsed router is a global minimum of a JSD term in isolation; measured against five random directions of matched norm, descending $\mathcal{L}_{rc}$ in fact *disperses* routing, seven times more than a random step.

The selection rule did not discriminate. We pre-registered the following rule: *adopt the largest* λ *within* 1.0% *of the control*. Every value passes, and so does every other arm in the paper: the fourteen non-divergent checkpoints span 0.06% on the fixed evaluation set, so the rule selects λ=0.20 only because "largest" is a tiebreak. The coefficient was therefore selected by the pre-registered tiebreak, not by a discriminating validation-loss comparison; the axis decomposition supports the interpretation of the selected arm, not its selection. The adopted arm also averages 1.9 inactive experts at layer 7 over the last ten logged points and has Gini $+0.078$, which the rule does not price, while λ=0.05 carries neither for indistinguishable loss. The full λ grid is in Appendix A; Figure 5 above shows where each constraint attaches.

D GENERATION: SAMPLER, METRIC SUITE AND PER-ARM RESULTS

The model has no `generate`, so we sample from it as a Markov random field (Wang & Cho, 2019), a procedure whose original justification was later shown not to hold exactly (Salazar et al., 2020). Each resampled position is masked: the fusion step excludes token t from its own prediction, but the neighbouring representations at $t-1$ and $t+1$ carry information from the original token t, so an unmasked chain largely returns each position to its current value (eight full sweeps moved GC only from 0.500 to 0.487). At T=0.8 the sampler scores **better** reconstruction loss than real hg38 (0.6342 against 0.8045). Relative to T=1.0 samples from the same sampler, which score 1.2587 nats, the T=0.8 samples have lower loss but $10.3\times$ the 3-mer Jensen–Shannon divergence from hg38

(0.09383 versus 0.009143); two disjoint hg38 samples of the same size give 0.00982, so the T=1.0 value is at this finite-sample floor and the T=0.8 value about 10× above it. **Selecting a sampler configuration by its same-model reconstruction score favours a setting with poorer genomic sequence statistics.** We also assess mixing across initialisations rather than assume it: run from real sequence as well as from uniform, two differently initialised chains end about 4.7–5.6× closer to each other than either is to hg38. This is consistent with mixing towards a shared distribution, but does not establish convergence to stationarity (Appendix A).

The metric layer, validated on controls. We report coverage (Naeem et al., 2020) with precision, recall (Sajjadi et al., 2018; Kynkäänniemi et al., 2019) and FID (Heusel et al., 2017) in two feature spaces, neither of them our own model's hidden states, since averaged samples look especially typical in the space that produced them. On controls alone the suite discriminates: a 3rd-order Markov chain shows a high-precision, low-recall signature on the metric-validation controls (precision 0.953, recall 0.000), the exact shape of the averaging failure reported for genomic generators (Appendix A).

The arms. 512 samples of 4,096 bp per arm at 8, 16 and 32 sweeps, scored against 328 held-out windows, with coverage stratified on the real side only so no classifier for generated sequence is required. **No arm separates from the control**: the largest between-arm coverage difference is 0.012 in the learned space and 0.052 in the k-mer space, against within-arm variation across sweep counts of up to 0.095 and 0.146 — the arms differ by less than the same arm differs from itself when its chain runs longer. Linear probes on frozen features say the same on three tasks, the largest gap being +0.013 MCC against a bootstrap standard deviation of 0.011. Held-out loss is the one endpoint where the arms do separate, and §4.2 explains why that does not license attribution.

Two facts hold regardless. All arms clear the Markov control in the learned space but the chain *beats* all three in k-mer space, worth stating for anyone evaluating a genomic generator on k-mer statistics; and every arm's recall collapses against real sequence, which is the averaging failure this suite was built to detect.

Variant effect: too little signal at this scale to judge. We also scored 2,000 ClinVar variants zero-shot, with the set and the reading rules fixed before any arm was scored. The control reads AUROC 0.522 (95% CI 0.497–0.547), which the pre-registered rule classes as uninformative, so we draw no conclusion about the arms from it. The direction is the one this paper would predict — the arms sit within a paired interval of the control while every routing intervention lowers AUROC with an interval excluding zero — but on a baseline this close to chance we do not read that as evidence (Appendix G).

E BLOCK-LEVEL ROUTING: FIVE ATTEMPTS

Routing on pooled blocks rather than tokens is a natural proposal and we could not make it work in five attempts: a gradient-checkpointing conflict; pooling the gate's *input*, which reached validation loss 15.1 with 25 of 64 experts dead; pooling probabilities, 1.700; the same at a reduced auxiliary coefficient, which recovered to 1.090 before diverging to 1.780; and a centred rule described below. Entropy stayed at 0.94–0.99 throughout while dead experts oscillated between 0 and 4, so §3's warning applies here too — entropy alone would have read this as healthy.

We expected pooling 128 hidden states to shrink between-block contrast by $\sqrt{128} \approx 11.3$; it shrinks by 2.7. Read as an effective routing correlation length, this gives 17–18 tokens, but that is not a property of the model: repeated at blocks 16 to 256 the implied length runs 4.5 to 26.9, growing as $b^{0.64}$, and a quantity that changes 6× across the range over which it is supposed to be constant is not a length scale. The source is in the same measurement — contrast shrinks as $b^{0.18}$, far short of the $\sqrt{b}$ independent samples would give, and the "k independent draws per block" picture that produced the estimate assumes exactly the $\sqrt{b}$ it does not obey. What the exponent says instead is that **routing correlation persists across every scale we measured** rather than dying at some length.

Mean-logit pooling is worse than mean-probability at every measured block size; switching between these rules alone did not yield stable training. Centring by subtracting the per-expert block mean holds the mode-pair share at 8–9% across a sixteen-fold range where mean-probability degrades from 15.5% to 33.5%. That transfers zero-shot and fails in training, in a new way: the centred arm's

first validation is **0.959**, nearer the control than any other block-routing arm reached at any point, and it then diverges to **9.16** by step 4,749 *with zero dead experts at every layer throughout*. The pre-registered router-health criterion would not have flagged this failure: it monitors expert activity, not the language-model loss, and was not implemented in the live training loop. The run was also mis-configured, its auxiliary coefficient derived from an inflation that centring removes, so we re-ran it elevenfold higher: the first three validations move by less than 0.13. Thus, this coefficient mismatch does not explain the early trajectory, although the corrected run does not establish long-run stability. Restoring between-block contrast removes the large initial loss penalty but did not prevent subsequent divergence in the original centred run. In these tested configurations, changing the pooling rule did not yield stable training, and correcting the coefficient did not materially change the early validation trajectory. Expert death and a large initial loss penalty are not necessary for the observed divergence; its mechanism remains undetermined.

F THE SMOOTHNESS TERM

Arm A2 adds

$$\mathcal{L}_{sm} = \frac{1}{B-1}\sum_{b=1}^{B-1} h_\delta(\|p_{b+1} - p_b\|_1)\,, \qquad h_\delta(d) = \begin{cases} d^2/2\delta & d < \delta \\ d - \delta/2 & d \geq \delta \end{cases}$$

over block-pooled routing distributions p_b, with $\delta = 0.3$ and $\lambda_{sm} = 10^{-3}$. Plain total variation would charge every transition equally and flatten the boundaries the routing is supposed to track; the Huber form charges frequent small wobble quadratically and leaves rare large transitions at linear cost. The weight was set the same way as λ_{rc} — by gradient share at the router, where it contributes 0.59% of the language model's — and not by loss share. It is a regulariser on block-to-block routing variance, not a claim about biological contiguity; enhancers and their targets plainly are not contiguous.

G DECOMPOSING THE PERMUTATION, AND A VARIANT-EFFECT TEST

Both measurements were registered before they were run, with reading rules and abort conditions; everything reported below passed the aborts.

Arms. Each replaces the MoE block's forward at inference, on the same forward-direction routers, windows and loss as the permutation table of §5; the model files are unchanged. For a token whose top-2 selection is S with weights w, and a logit permutation π: **P** sends it to $\pi^{-1}(S)$ with weights w, which is what permuting the logits does; **P-gate** sends it to $\pi^{-1}(S)$ with the router's own probabilities for those experts, rescaled to the original pair's total mass (JanusDNA does not renormalise its top-2 weights, so that total carries information); **P-gate-raw** omits the rescaling; **Swap** applies one transposition per layer, either the pair of experts with the most similar utilisation or a random pair (five draws); **Flip** keeps S and exchanges the two weights. P and P-gate use five permutations each. An identity arm, which replaces the forward and changes nothing, reproduces the unpatched loss to 10^{-5} nats on every checkpoint, and P reproduces the permutation costs of §5 to the second decimal — both registered abort conditions.

JanusDNA, $4 \times 131{,}072$ **bp.** Loss change, share of (token, slot) pairs whose expert changed, cost per rerouted slot, and — the normalisation added after the Genos Flip result, so post hoc — routing mass displaced, the total variation between the original and edited per-expert allocations, with the cost per unit of it.

checkpoint	arm	Δ loss	rerouted	per slot	mass moved	per unit mass†
released	P	$+7.34 \pm 0.84\%$	97.5%	7.5%	0.310	23.7%
	P-gate	$+6.59 \pm 0.38\%$	97.5%	6.8%	0.305	21.6%
	P-gate-raw	$+5.69 \pm 0.37\%$	97.4%	5.8%	0.193	29.5%
	Swap, closest	$+0.68\%$	12.6%	5.4%	0.033	20.4%
	Swap, random	$+0.86 \pm 0.31\%$	12.0%	7.2%	0.042	20.2%
	Flip	$+1.59\%$	0	—	0.090	17.6%
A0	P	$+7.43 \pm 1.63\%$	97.4%	7.6%	0.291	25.5%
	P-gate	$+6.64 \pm 0.56\%$	97.5%	6.8%	0.287	23.1%
	P-gate-raw	$+5.85 \pm 0.38\%$	97.4%	6.0%	0.185	31.6%
	Swap, closest	$+1.31\%$	15.7%	8.4%	0.057	23.1%
	Swap, random	$+0.82 \pm 0.27\%$	12.2%	6.7%	0.040	20.6%
	Flip	$+1.50\%$	0	—	0.079	19.0%
A4nb0	P	$+6.45 \pm 0.55\%$	98.2%	6.6%	0.241	26.8%
	P-gate	$+5.86 \pm 0.44\%$	98.2%	6.0%	0.239	24.5%
	P-gate-raw	$+5.50 \pm 0.40\%$	98.1%	5.6%	0.168	32.8%
	Swap, closest	$+1.04\%$	14.5%	7.2%	0.039	27.0%
	Swap, random	$+0.73 \pm 0.17\%$	11.6%	6.2%	0.030	24.3%
	Flip	$+0.99\%$	0	—	0.050	19.8%

† Post hoc. P-gate-raw also changes the total mass of the pair, which the displacement measure does not separate from moving it; it is the one arm outside the 18–27% range and is excluded from that statement.

Against the registered rules: P-gate is at least 5% and at least two thirds of P on the control (6.64%, 89%), so the correspondence carries the cost; Swap's cost per rerouted slot is within a factor of two of P's on every checkpoint (0.72–$1.10\times$ for the closest pair, 0.88–$0.95\times$ for random pairs); and Flip is well below P (15–22% of it). Swapping the pair with the most similar utilisation is not special: random pairs cost the same per slot.

Genos-1.2B, $16 \times 8{,}192$ tokens, all twelve MoE layers. Mixtral renormalises its top-2 weights, so P-gate is mass-preserving by construction and P-gate-raw has no counterpart.

arm	Δ loss	rerouted	per slot	mass moved	per unit mass†
P	$+134.5 \pm 4.0\%$	82.2%	164%	0.749	180%
P-gate	$+126.6 \pm 9.2\%$	82.4%	154%	0.698	181%
Swap, closest	$+82.2\%$	24.9%	330%	0.234	351%
Swap, random	$+87.0 \pm 4.6\%$	25.2%	345%	0.239	364%
Flip	$+115.5\%$	0	—	0.509	227%

P-gate retains 94% of P. The registered per-slot rule sits on its boundary here: a single swap costs 2.0–$2.1\times$ as much per rerouted slot as the full permutation. One candidate is a ceiling on the full permutation, which already equals random routing (134.9%), so that the per-slot figure for P is depressed; we did not measure a floor and do not claim it. Flip costs 86% of P, which under the registered rule means that on Genos the weighting matters as much as the choice. The mass measure says why: Genos' renormalised pair is peaked, and exchanging its weights moves 0.51 of the routing mass onto the second-ranked expert, a displacement two thirds the size of a full permutation's.

ClinVar zero-shot. 2,000 single-nucleotide variants from the GRCh38 release dated 2026-09-13 — $1{,}000$ pathogenic or likely pathogenic and $1{,}000$ benign or likely benign, review status of at least one star, chromosomes 1–22 and X, reference base verified against the assembly — drawn with a fixed seed. Score: $\log p(\mathrm{ALT}) - \log p(\mathrm{REF})$ at the centre of an $8{,}192$ bp window with that position masked, restricted to the four bases. AUROC takes pathogenic as positive; intervals are from $1{,}000$ bootstrap resamples of variants, the same resamples for every condition, so the differences from A0 are paired. Coding covers missense, synonymous, nonsense, stop-loss and start-loss variants ($1{,}412$); the split is reported, not used to select.

condition	AUROC (95% CI)	Δ vs A0 (95% CI)	coding	non-coding
released	$0.519\ [0.492, 0.544]$	$[-0.008, +0.000]$	0.505	0.554
A0 (control)	$0.522\ [0.497, 0.547]$	—	0.508	0.560
A2	$0.524\ [0.498, 0.549]$	$[-0.001, +0.005]$	0.510	0.563
A4nb	$0.521\ [0.495, 0.546]$	$[-0.004, +0.002]$	0.508	0.557
A4nb0	$0.520\ [0.493, 0.545]$	$[-0.006, +0.001]$	0.505	0.559
A5nb	$0.521\ [0.495, 0.547]$	$[-0.004, +0.003]$	0.508	0.558
A6	$0.520\ [0.494, 0.545]$	$[-0.006, +0.001]$	0.505	0.558
A6h	$0.518\ [0.492, 0.543]$	$[-0.008, -0.001]$	0.503	0.557
A0 + P, three permutations	0.495–0.500	all exclude 0	0.487–0.491	0.496–0.536
A0 + P-gate, three	0.497–0.500	all exclude 0	0.485–0.491	0.510–0.536
A0 + random routing, two	0.500–0.504	all exclude 0	0.490–0.497	0.523–0.528

The registered rule for a control AUROC at or below 0.55 is that the task cannot separate arms, and we apply it. None of the fifteen exploratory variant-effect comparisons was multiplicity-corrected; A6h is the lone nominal difference whose interval excludes zero (by 0.001), and we do not interpret it as evidence of intervention benefit or harm. What little signal there is lies in the non-coding variants (0.560 on the control); promoting that subset after seeing it would be the selection the rule exists to prevent. The window is shorter than the 131 kbp the model was trained on, fixed for cost before scoring, and whether a longer context carries more signal is not addressed here.

H Design implications and open questions

This appendix states what the paper implies for anyone building a genomic MoE, and marks each implication by the evidence behind it. Some follow directly from measurements in the main text; some depend on a regime this model is in and the two others are not; and some are hypotheses the results suggest and do not test. We keep the three apart because our record of corrections (supplementary material) shows that the gap between a natural reading and a measured one is where most of our errors lived.

Measurement limits

The structural statistics recommended in §8 are selected-checkpoint measurements, not training-log statistics or validated downstream surrogates. On JanusDNA, one sampled expert-pair swap reroutes 12–16% of slots and costs 6.7–8.4% per rerouted slot, close to the full permutation's 7.6%; on Genos-1.2B a single swap costs 2.0–$2.1\times$ as much per rerouted slot (Appendix G), so sampled swaps should be calibrated against at least one full permutation. Class enrichment requires annotated sequence and a within-window matched null, and effective rank requires evaluating all experts on shared inputs. No general warning threshold is established; trends should be compared on a fixed evaluation set.

H1: balance early, then anneal

The auxiliary objective exists to prevent collapse, and its value is highest before every expert has found traffic. Once none is dead and single-expert ablations are cheap, as in JanusDNA, continuing to regulate the histogram moves a quantity to which validation loss is comparatively insensitive (§4), while arms with a per-layer prior-targeted term show lower layer-7 permutation sensitivity by about 39% (§5). This suggests annealing the balancing coefficient once the dead-expert count reaches zero and the structural statistics stabilise. The hypothesis is conditional on the regime: PlantBiMoE and Genos-1.2B rely heavily on single experts (§6), and our data do not establish whether balancing caused or prevented that concentration. A0z is consistent with H1 but does not test it, because it was fine-tuned from a checkpoint pretrained with balancing. The schedule, its trigger and its effect from initialisation are untested.

Which assignments carry which function

§5 establishes that the assignment matters in aggregate. Whether a class-enriched expert's assignment carries that class's function is a different question, and the released harness could test this at inference

implication	evidence in this paper	status
compute and monitor routing per layer	pooled Gini 0.0068 against 0.164–0.407 per layer (§4)	measured
report Gini and dead experts alongside entropy	layer-7 entropy 0.937 with block-pooled Gini 0.696 (§3); RC arm at λ=0.20 averages 1.9 inactive layer-7 experts (§7)	measured
evaluate assignment-level statistics at selected checkpoints	A0 permutation +7.48% (128 windows); A4nb0 L7 enrichment and rank below A0 (§5)	measured on fixed checkpoints
calibrate by gradient share at the router	shipped λ would be 38× the LM gradient (§4)	measured
decompose a symmetry before constraining it	complement fails, reversal survives (§7, App. C)	measured
do not select a generator by its own reconstruction score	0.6342 against hg38's 0.8045 (App. D)	measured
consider relaxing balancing only after checking expert reliance	worst ablation JanusDNA 0.27%; PlantBiMoE 83.4%; Genos 73.0% (§6)	conditional implication
balance early, then anneal	consistent with A0z; no schedule trained	hypothesis
measure biological structure before imposing it	regret objective optimised its proxy (App. B)	hypothesis

Table 7: Design implications, the evidence each rests on, and how far that evidence reaches.

using the interventions already implemented: select expert–class pairs whose enrichment exceeds the matched null's 99th percentile; swap that expert's identity with a non-enriched expert of matched utilisation at the same layer, which leaves the histogram unchanged up to relabelling as every permutation here does; and measure the loss increase separately on tokens of the enriched class and on all other tokens, against the same swap between two non-enriched experts of matched utilisation. If the assignment carries class function the increase concentrates on the enriched class; if it does not, the value of the assignment is distributed rather than class-aligned, which is itself informative about what enrichment statistics can be trusted to mean. The observed pair-swap costs are 0.82–1.31% on the four-window protocol; the proposed class-stratified comparison would require its own paired uncertainty estimates. **We state this rather than report it because the reading rules were not registered before the enrichment table was seen**, and a class-by-class comparison chosen after that table is the selection our pre-registration exists to prevent; it is the first thing we would register next.

Open questions

When to stop. The paper's conclusion is that the question is when and how to adjust balancing, not whether to discard it; H1 proposes a trigger, and none has been tested. *Specialisation or redundancy.* The fall in permutation sensitivity under per-layer balancing reads either as the assignment carrying less or as the layer becoming more robust; separating the two needs an endpoint that rewards redundancy, such as degradation under expert ablation, input noise or distribution shift. *From scratch and across genomes.* Every trained arm here is fine-tuned from one 35.6 M checkpoint on one species; the layer-7 reduction in particular needs replication from initialisation and on other genomes. *E and k.* Three models give three points in a space the diagnostics depend on; a controlled sweep over E and k at fixed backbone and data would say how much of the regime difference in §5 is architecture and how much is training.

I Measurement and protocol details

Samples. Reverse-complement agreement uses 64 non-overlapping 131 kbp windows with a default block size of 128 bp (128 tokens at single-nucleotide resolution). Other measurements use separate samples, identified where reported: four windows for the released-checkpoint Gini comparison, eight for arm-level Gini, and 128 held-out windows for JanusDNA validation loss and the headline permutation estimate.

Two nulls for routing agreement. For a transformation T, we compare the top-k experts selected for each block with those selected for its transformed counterpart and chance-correct the observed overlap o as $\kappa = (o-n)/(1-n)$, where n is the overlap expected under a null. The *uniform* null uses the overlap expected under uniform expert assignment. The *matched* null breaks the correspondence between each source block and its transformed counterpart by shuffling transformed blocks within the same window, while preserving each router's marginal expert usage. The distinction matters when marginal usage is non-uniform, as it is here (token-level utilisation Gini 0.164–0.407 across the four MoE layers, four windows). As routing is averaged over larger blocks, each block's aggregate routing moves toward the same marginal distribution, so unrelated blocks increasingly select the same heavily used experts. In a simulation with *zero* shared correspondence but the same non-uniform marginal usage, κ under the uniform null rises from 0.058 to 0.953 as block size increases from 1 to 512 bp, while under the matched null it remains within ± 0.03 of zero. Three patterns that appeared meaningful in our early analysis disappeared under the matched null (supplementary material).

Entropy alone can miss routing collapse. At the released checkpoint, with routing pooled over 128 bp blocks, normalised gating entropy ranges from 0.937 to 0.995 across the four MoE layers, yet block-pooled utilisation Gini reaches 0.696 at layer 7, where entropy is still 0.937, with no dead experts in any layer: a nearly flat softmax can still contain a small, persistent bias that repeatedly sends blocks to the same experts. The concentration depends on the aggregation scale; at layer 7, Gini increases from 0.490 at block size 1 to 0.571, 0.696 and 0.756 at block sizes 16, 128 and 512. For comparison, reinitialising only the routers on top of the released representations gives normalised entropy 0.932–0.998 with no dead experts. We report utilisation Gini and dead-expert counts alongside entropy.

PlantBiMoE evaluation. We sample 32 windows of 512 bp from TAIR10 chromosome 1 using uniformly sampled start positions (NumPy seed 0), retaining windows with less than 1% `N` bases. Single-nucleotide tokenisation adds two special tokens, giving input shape $(32, 514)$. We replace each input position with `[MASK]` independently with probability 0.15 (seed 0), including the special-token positions, and compute cross-entropy only on masked positions. The same windows and mask are used for the baseline, five permutations, five random-routing draws, and one ablation for each of the four experts. The changed-slot fraction for permutation was not logged. The unmodified masked-input baseline on this sample is 1.2861 nats. An initial evaluation without masking gave a baseline of only 0.0047 nats, because input and target were visible at the same position, and inflated the permutation comparison to $+7256\%$; every reported PlantBiMoE result therefore uses the fixed mask. There is no separate train/validation split or verified exclusion of pretraining sequences: these numbers measure paired inference-time intervention sensitivity on the sampled inputs, not held-out predictive performance. The checkpoint configuration (`num_experts=4`, `num_experts_per_tok=1`) overrides the implementation defaults of 16 experts and top-2 routing.

Genos-1.2B evaluation. Interventions use $16 \times 8{,}192$ tokens over all twelve MoE layers (Appendix G). Evaluation context matters: the permutation penalty is about 56% at 2,048 tokens and 134% at 8,192.

An inference-time bias does not isolate utilisation. Our trained JanusDNA interventions change the utilisation target and let the router adapt during fine-tuning. A post hoc router bias cannot reproduce this comparison while holding token–expert assignment fixed, because changing hard-assignment counts necessarily changes which experts receive some tokens. In a router-bias sweep on Genos-1.2B (Appendix A), even the smallest tested bias spreads the histogram $1.41\times$, changes the selected expert set for 16.9% of tokens, and raises evaluation loss by 47.4%; larger biases change more assignments and produce larger losses. Because utilisation and assignment change together, this experiment cannot attribute the loss increase to the histogram alone, and the 0.06% span of the fine-tuned JanusDNA checkpoints is not directly comparable with it.

A0z and the scope of the balancing claim. A0z starts from the released JanusDNA checkpoint, which was pretrained with balancing, and disables the shipped balancing term only during fine-tuning. All 16 experts in each MoE layer remain active under the measured conditions, and its layer-7 permutation penalty (2.23%) lies between the released checkpoint (2.18%) and A0 (2.35%). A0z therefore shows that this pretrained router can remain active without the auxiliary term during fine-

tuning, not that a router trained from scratch would do so. In PlantBiMoE and Genos-1.2B, removing the most consequential expert raises evaluation loss by 83.4% and 73.0%; these results show that individual experts can be critical, but not whether balancing caused or prevented that concentration.

J BALANCING-COEFFICIENT CALIBRATION

Pooling hides layer-wise imbalance. At the released checkpoint the pooled Gini (0.0068) is 24–$60\times$ smaller than the layer-wise values (0.164–0.407). Three measurements show that this mismatch is not explained by an auxiliary coefficient that is simply too small. First, at identical routing, computing the objective per layer gives a router-gradient magnitude of 73.4, against 1.79 for the pooled form, a $41\times$ difference; much of the layer-wise imbalance signal cancels before reaching the optimiser. Second, in a separate coefficient-sweep experiment (not the released-checkpoint or A0 Gini evaluation), increasing `router_aux_loss_coef` from 0.2 to 2.0 raised layer-7 Gini from 0.454 to 0.516 and mean per-layer Gini from 0.290 to 0.321; the test was registered with a null prediction under the grouping explanation, and the observed change moved toward greater imbalance. Third, the released balancing term already contributes 36% of the language-modelling gradient magnitude at the router. Genos independently reports a balanced-load mechanism for imbalance its authors associate with the four-base vocabulary (Lin et al., 2025a); our intervention targets a different issue, how the objective is aggregated across layers.

Per-layer objective and coefficients. When $t_e = 1/E$, each layer's term of $\mathcal{L}_{\text{bal}}$ (§4) reduces to the standard uniform-target objective; we verify this equivalence numerically. The coefficient cannot be copied from the pooled objective. At $\lambda = 1$, the pooled uniform objective produces a router gradient $1.79\times$ that of the language-modelling objective, whereas the per-layer uniform and per-layer prior objectives reach $73.4\times$ and $189.7\times$. Reusing the released coefficient of 0.2 would make the per-layer prior balancing gradient about $38\times$ the language-modelling gradient at the router. We instead match the router-gradient share of the released term, giving $\lambda_{\text{bal}} = 0.0049$ for the per-layer uniform form and 0.0019 for the per-layer prior form; these 40–$100\times$ reductions reflect the change in gradient scale. The per-layer uniform coefficient ($\lambda = 0.0049$) is reported as a calibration reference; no trained arm used this coefficient. The trained prior-targeted arms use $\lambda = 0.0019$.