Ares: Loss-Free Mixture-of-Experts Routing for Bidirectional Protein Encoders

Hazem Alsamkary [1]

Abstract

Scaling protein language models has predominantly relied on increasing dense transformer parameters, while loss-free MoE routing strategies naturally suited to bidirectional architectures, such as Soft-MoE and Expert-Choice, remain unexplored in protein encoders. We bring both routing strategies to protein encoders and systematically compare them under identical compute budgets. We find that Soft-MoE trains stably, while Expert-Choice exhibits cascading routing instability that forces early termination when MoE layers are stacked consecutively and produces recurrent transient collapses when interleaved with dense layers. Despite training on only 39B tokens, roughly 1/25th the budget of ESM-2, our Soft-MoE variants match or exceed ESM-2 3B on Fluorescence and GB1 and ESM-2 650M on Stability, while trailing on Remote Homology, Secondary Structure, and ProteinGym by margins consistent with the token gap. Within Soft-MoE, ablating router-side L_2 normalization reveals that L_2 functionally disables the dispatch step yet matches the unconstrained variant on downstream tasks, suggesting dispatch contributes less to representation quality at this scale than the standard Soft-MoE framing predicts. Together, these results position continuous routing as a more robust foundation than discrete top-k for MoE-based protein encoders.

1. Introduction

Protein language models have emerged as powerful tools for understanding protein structure and function, learning rich representations from large-scale sequence databases without explicit structural supervision (Hayes et al., 2025; Nijkamp et al., 2022; Bhatnagar et al., 2025; Elnaggar et al., 2023). Models such as ESM-3, ESM-2, ProGen2, ProGen3, ProtTrans, and Ankh have demonstrated that scaling transformer architectures and training data leads to consistent improvements across downstream tasks, including structure prediction, fitness prediction, and function annotation. However, scaling these models has predominantly followed a single strategy: increasing the size of dense transformers, where every parameter is active for every input token. This approach ties model capacity directly to computational cost, making further scaling increasingly expensive.

Mixture-of-Experts offers an alternative scaling paradigm, decoupling total model capacity from per-token compute by routing each token to a subset of specialized expert networks (Shazeer et al., 2017). MoE has been successfully adopted in natural language processing and vision (Fedus et al., 2022; Riquelme et al., 2021), yet remains largely unexplored in protein language modeling. The only notable exception is ProGen3, which explored MoE in autoregressive protein generators (Bhatnagar et al., 2025). However, the dominant protein language models are bidirectional encoders, and a family of routing strategies, Soft-MoE and Expert-Choice routing, are naturally suited to non-autoregressive architectures (Puigcerver et al., 2024; Zhou et al., 2022). We review both in Section 2. Despite their natural fit for protein encoders, these approaches have not been investigated in this domain.

In this work, we bring Soft-MoE and Expert-Choice routing to bidirectional protein encoders for the first time. We train four 4B-parameter MoE models on UniRef50 using TPU pods under identical compute budgets, systematically comparing routing strategies and MoE layer placement, interleaved versus consecutive.

Our contributions are as follows. First, we present the first application of loss-free MoE routing in protein language models, providing a controlled comparison between Soft-MoE and Expert-Choice under identical data, compute, and evaluation conditions. Second, we analyze the effect of MoE layer placement on training stability and identify a placement-driven asymmetry between continuous and discrete routing. We argue that this asymmetry follows from applying a known property of top-k routing, the gradient mismatch introduced by piecewise-constant selection under a straight-through estimator, to the depth dimension; the contribution is the empirical observation that this property

[1]Proteinea. Correspondence to: Hazem Alsamkary <hazemalsamkary@gmail.com, hazem@proteinea.com>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

matters in deep bidirectional protein encoders. We evaluate our models across a diverse set of structure and function benchmarks. We release our full codebase including training infrastructure, model weights, and evaluation pipelines[1] to facilitate future research on MoE protein language models.

2. Background

2.1. Mixture-of-Experts and the Load-Balancing Problem

Mixture-of-Experts replaces a single feed-forward layer with a set of E expert networks and a router that dispatches each input token to a subset of them (Shazeer et al., 2017). Only the selected experts are executed for a given token, so total parameter count grows with E while per-token compute remains approximately constant. Modern transformer MoE layers follow this template (Lepikhin et al., 2020; Fedus et al., 2022), typically scoring each token against every expert with a learned router and selecting the top-k experts per token, with k as small as one in Switch Transformer (Fedus et al., 2022).

This token-choice formulation has a structural load-balancing problem. Without intervention, the router tends to concentrate tokens on a small number of experts, leaving the rest undertrained and wasting the capacity that MoE is meant to unlock (Shazeer et al., 2017; Fedus et al., 2022). The standard mitigation is an auxiliary balancing loss added to the training objective, which penalizes imbalanced routing and encourages uniform expert utilization. Auxiliary losses trade one problem for another: they introduce a hyperparameter that must be tuned against the main loss, and the penalty pressure can conflict with what would otherwise be the router's learned specialization. A separate issue is token dropping: when more tokens are assigned to an expert than its capacity allows, the overflow is dropped from the computation, which is particularly costly in settings where every token carries a supervised signal.

2.2. Loss-Free Routing

Two routing strategies address these issues without an auxiliary balancing loss. We refer to them collectively as *loss-free* routing in the sense that neither requires an auxiliary balancing loss term in the training objective.

Soft-MoE (Puigcerver et al., 2024) replaces discrete token-to-expert assignment with a fully differentiable soft assignment. Rather than routing each token to a single expert, each expert receives a set of *slots*, where each slot is a weighted combination of all input tokens. The per-expert computation runs on these slot representations, and the outputs are combined back into token representations via a second set of learned weights. Because every token contributes to every slot and every expert output contributes to every token, there is no discrete selection, no token dropping, and no imbalance to correct for. The authors report that this formulation stabilizes training and scales to large numbers of experts without the instabilities observed in top-k MoEs.

Expert-Choice (Zhou et al., 2022) inverts the standard routing direction. Instead of each token selecting its top-k experts, each expert selects its top-k tokens from the batch according to a learned routing score. Load balancing is then guaranteed by construction: every expert processes exactly the same number of tokens, and no auxiliary loss is needed to enforce balance. A token can be selected by multiple experts, by one, or by none, so the allocation of compute across tokens is heterogeneous, tokens the router finds informative are processed by more experts than tokens it does not. In our Expert Choice implementation, routing scores are normalized with a softmax over the token dimension rather than over the expert dimension as in the original formulation (Zhou et al., 2022). Consequently, our results should be interpreted as evaluating this token normalized variant of Expert Choice rather than the canonical formulation. We do not evaluate the original expert-axis normalization in this work.

Both strategies were originally developed and evaluated in non-autoregressive settings, Soft-MoE in vision transformers and Expert-Choice in bidirectional language encoders, where attending to the full set of tokens at routing time is permissible. This makes them a natural fit for bidirectional protein encoders, which are the setting we study in this work. Other recent approaches to balancing without an auxiliary loss include bias-based balancing on top of top-k token choice (DeepSeek-AI, 2024) and softer token-side aggregations such as Mixture-of-Tokens (Antoniak et al., 2024); we do not evaluate these alternatives, and focus on the Soft-MoE versus Expert-Choice contrast.

3. Method

3.1. Architecture

Our base architecture is a 20-layer, $d = 1024$ encoder-only transformer with bias-free linears, SwiGLU FFNs (intermediate dimension 4096), Grouped-Query Attention (Ainslie et al., 2023) with 16 query / 8 key-value heads, pre-norm RMSNorm (Zhang & Sennrich, 2019), and Rotary Position Embeddings (Su et al., 2023) (base 10000). In total, the model comprises approximately 4.2B parameters across the encoder, distributed across both attention and MoE expert networks. Active parameters per token are approximately 4.2B for Soft-MoE (every expert executes on its slots) and approximately 440M on average for Expert-Choice at ca-

[1]Code: `https://github.com/hazemessamm/ares`. Weights: `https://huggingface.co/hazemessam`.

pacity factor 2.0; Zhou et al. (2022) set $c = 2$ to match the aggregate per-token compute of token-choice routing with two experts activated per token, and we adopt the same convention.

We study two MoE layer placement strategies. The *interleaved* configuration alternates MoE and dense FFN layers (odd-indexed layers are MoE); the *consecutive* configuration places 10 dense layers followed by 10 MoE layers. Both use 32 experts per MoE layer. The four trained variants are: Expert-Choice consecutive, Expert-Choice interleaved, Soft-MoE consecutive with router L_2 normalization, and Soft-MoE consecutive without router L_2 normalization (the latter two form the ablation in Section 3.2). We do not train Soft-MoE interleaved: the structural argument in Section 3.3 predicts that Soft-MoE is placement-invariant, so a Soft-MoE interleaved run would not be informative about routing behavior, and the matched-compute budget was better allocated to the L_2 ablation.

3.2. Routing Strategies

We apply the two routing strategies reviewed in Section 2.2, Soft-MoE and Expert-Choice, to our encoder. Below we describe the specific hyperparameters used for each and, for Soft-MoE, an ablation of the router-side L_2 normalization from the original formulation.

Soft-MoE (Puigcerver et al., 2024). We use 64 slots per expert (32 experts per MoE layer) with a noise level of 0.05.

The original Soft-MoE formulation applies L_2 normalization to both the input tokens and the slot parameters before computing routing logits, with a learnable scalar rescaling the normalized slots. This makes the routing computation a scaled cosine similarity between tokens and slots, removing magnitude information from both sides; the original motivation was stability concerns observed in very deep vision transformers (Puigcerver et al., 2024). Our encoder already applies RMSNorm in a pre-norm configuration before every attention and feed-forward block, which constrains the scale of the inputs to the routing function. We therefore ablate the router-side L_2 normalization directly, training two otherwise-identical Soft-MoE consecutive variants, one with L_2-normalized routing (as in the original formulation) and one without. The ablation is motivated by two hypotheses: (i) that the model's existing pre-norm architecture is sufficient to stabilize routing without an additional router-side normalization, and (ii) that allowing unnormalized tokens and slots preserves magnitude information that may be useful for expert specialization, since the relative magnitude of inner products, not only their direction, can carry a signal about token-slot affinity. We report the downstream consequences of this ablation in Section 4.1 and in the benchmark results.

Expert-Choice (Zhou et al., 2022). We use a capacity factor of 2.0 and a noise level of 0.05.

3.3. Routing Continuity and Layer Placement

The two routing strategies differ in a structural property that, we argue, is responsible for the stability behavior observed empirically in Section 4.1. We do not claim a new theoretical result here, but rather observe that a known property of each routing operation has a direct consequence for how these layers behave when stacked consecutively.

Soft-MoE dispatch is continuous. Given tokens $X \in \mathbb{R}^{m \times d}$ and slot parameters $\Phi \in \mathbb{R}^{d \times p}$, Soft-MoE computes logits $L = X\Phi$, a dispatch matrix $D = \mathrm{softmax}_{\text{col}}(L)$, and a combine matrix $C = \mathrm{softmax}_{\text{row}}(L)$ (Puigcerver et al., 2024). The layer output is a composition of softmax, matrix multiplication, and the per-expert MLPs. Softmax and matrix multiplication are C^∞, and our per-expert MLPs are SwiGLU and therefore also C^∞ in their inputs and parameters; the dispatch weights are everywhere positive. Puigcerver et al. note that all operations in Soft-MoE layers are continuous and fully differentiable (Puigcerver et al., 2024, Section 2.2), and that the mechanism is immune to token dropping and expert imbalance by construction, since every slot is a weighted average of all tokens (Puigcerver et al., 2024, Section 2.2). They further demonstrate scaling to thousands of experts without the quality degradation observed in sparse baselines (Puigcerver et al., 2024, Figure 6). Stacking k Soft-MoE layers consecutively yields a composition of smooth functions, which is itself smooth. Therefore, there is no structural reason to expect layer placement to affect training stability for Soft-MoE.

Expert-Choice dispatch is piecewise constant. Given the same logits L, Expert-Choice constructs a dispatch indicator by selecting, for each expert, the top-k tokens with the highest routing scores (Zhou et al., 2022). The top-k operation is not continuous: an arbitrarily small update to Φ that changes the ordering of two tokens near the selection boundary produces a discrete change in the set of tokens dispatched to an expert, and hence a discrete change in the layer's output. Gradients through this selection are typically routed via a straight-through estimator, creating a mismatch between the forward computation (hard selection) and the backward computation (soft gradient). This gradient mismatch is a well-known property of top-k routing rather than a novel observation.

Cascading across consecutive layers. The consequence of this difference becomes visible when multiple MoE layers are stacked. In the consecutive configuration, 10 Expert-Choice layers are composed directly, with no intervening dense layers. A selection flip at layer ℓ changes the input distribution to layer $\ell + 1$, which can in turn cross its own selection boundaries, and so on. The same parameter up-

date thus has the potential to trigger correlated, discrete routing changes at multiple depths simultaneously. In the interleaved configuration, a dense feed-forward layer follows each Expert-Choice layer, applying a smooth, everywhere-defined transformation that partially absorbs upstream perturbations before they reach the next routing decision. This interleaving does not remove the underlying discontinuity; each Expert-Choice layer remains piecewise constant in its parameters, but it breaks the direct composition of discontinuous operations.

This structural view predicts three empirical regimes, all of which we observe in Section 4.1: (i) consecutive Expert-Choice should exhibit the most severe instability, since discrete routing flips compose directly across depth; (ii) interleaved Expert-Choice should be more stable than consecutive but still subject to occasional disruptions driven by individual routing flips; and (iii) Soft-MoE should be placement-invariant, since its dispatch is smooth regardless of how many MoE layers are stacked. The argument is structural; the empirical results below provide the quantitative evidence.

3.4. Training Recipe

Data. We train on UniRef50 (Suzek et al., 2007), comprising approximately 68 million protein sequences.

Truncation. We set the maximum sequence length to 1024 and randomly truncate longer sequences to a 1024 residue window. We treat the boundary tokens as markers of true protein termini rather than truncation boundaries: ¡cls¿ is prepended only when the retained window contains the native N-terminus, and ¡eos¿ appended only when it contains the native C-terminus, so a purely interior fragment receives neither. This keeps the boundary tokens aligned with genuine termini rather than arbitrary truncation cut points.

Sequence Packing. To our knowledge, no existing protein language model has reported using sequence packing during pretraining. Standard approaches pad each sequence to the maximum length, wasting compute on non-informative padding tokens. We implement offline sequence packing using a worst-fit-decreasing bin packing algorithm. Sequences are sorted by length in descending order, then greedily assigned to bins of maximum length 1024 using a max-heap that tracks remaining capacity per bin, always placing each sequence into the bin with the most remaining space. This reduces the dataset from approximately 68 million individual sequences to approximately 18 million packed bins, achieving a packing efficiency of 3.7x. During training, block-diagonal attention masks prevent cross-sequence attention within each packed bin, ensuring that packing is mathematically equivalent to training on individual sequences. All models are trained on identical packed data, ensuring that every model sees the same tokens in the same order with the same masking for each epoch.

Curriculum Masking. Rather than using a fixed masking probability, we employ a staged curriculum that gradually increases masking difficulty over training. We define four masking rates, 0.15, 0.20, 0.25, and 0.30, spanning the range from the BERT/ESM-2 default (Devlin et al., 2019) to the upper end of stable MLM training adopted by ModernBERT (Warner et al., 2024) and motivated by recent work showing that higher and dynamically scheduled masking rates can improve MLM pretraining (Wettig et al., 2023; Ankner et al., 2024). We sample among the four rates according to a time-varying probability distribution. Training begins with a probability distribution of $[1.0, 0.0, 0.0, 0.0]$, using only the lowest masking rate. Over training, each subsequent masking rate is linearly introduced at intervals of $\sim$30K steps, until all four rates are sampled uniformly with distribution $[0.25, 0.25, 0.25, 0.25]$ after approximately 120K steps. The remaining training proceeds with uniform sampling across all rates.

Optimization. We use AdamW with a peak learning rate of 3×10^{-4}, cosine decay schedule, and a warmup phase of 5% of total training steps. We apply weight decay of 0.01 and gradient clipping at 1.0. All models are trained in bfloat16 mixed precision.

Compute. All four model variants are trained on TPU v5e-64 / v6e-64 pod slices. The global batch size is 256 packed sequences per step, corresponding to 262,144 tokens per step. All variants are trained under identical compute budgets on the same data with the same masking schedules to ensure a fair and controlled comparison.

4. Results

4.1. Training Stability

The training stability results in this section were not planned in advance: they emerged as a sequence of observations and hypothesis-driven responses during training, which we report in chronological order. Our original intent was to train Soft-MoE and Expert-Choice in the same consecutive configuration (10 dense layers followed by 10 MoE layers) for a direct routing-strategy comparison. However, training Expert-Choice consecutive produced severe validation instability that forced us to terminate the run. We then hypothesized, based on the structural argument earlier formalized in Section 3.3, that the instability was driven by the direct composition of discontinuous routing operations across consecutive layers, and that interspersing dense layers between Expert-Choice layers should interrupt that composition. We tested this by training Expert-Choice with an interleaved placement. The resulting run was trainable through the full schedule but exhibited a different, milder failure mode: transient collapse-and-recover events that per-

sisted throughout training. We monitor masked language modeling validation accuracy at all four curriculum masking rates (0.15, 0.20, 0.25, 0.30); the patterns we describe below are consistent across all four rates, and we report the 0.15 curve as representative given that the rest of the rates follow the same pattern.

Expert-Choice consecutive diverges and cannot be recovered. Training proceeds smoothly for the first ∼80K steps, with validation accuracy climbing to approximately 0.28 by step 80K. At step 83K, the validation curve breaks into large-amplitude oscillations that grow in magnitude over the subsequent steps, with accuracy repeatedly dropping from ∼0.28 to below 0.18 within a few thousand steps and failing to stabilize. We terminated training at step 104K, after ∼21K steps of visibly degraded behavior, and retained the step-86K checkpoint for downstream analysis as the last checkpoint prior to severe oscillation. This run is shown in Figure 1 (left).

Expert-Choice interleaved remains trainable but exhibits transient collapses. The interleaved variant trains smoothly to ∼80K steps, reaching comparable pre-instability validation accuracy. Starting around step 85K, the validation curve begins to exhibit sharp, transient downward spikes, with individual points dropping from the running accuracy of ∼0.29 to values as low as 0.05 before recovering to the prior level within a handful of validation intervals. These collapse-and-recover events continue intermittently throughout training up to step 150K, including a second major drop near step 135K. Critically, the upper envelope of the curve is approximately maintained: the model is not diverging, but is experiencing recurrent routing disruptions from which it recovers. This run is shown in Figure 1 (right).

Soft-MoE trains stably, with and without router L_2 normalization. Under identical data, curriculum, and compute conditions, both Soft-MoE consecutive variants train without exhibiting either the divergent oscillations of Expert-Choice consecutive or the recurrent transient collapses of Expert-Choice interleaved. The L_2 variant trains monotonically; the no-L_2 variant exhibits one isolated transient drop near step 140K from which the model recovers within a few thousand steps, qualitatively distinct from the recurrent collapse-and-recover events that persist throughout Expert-Choice interleaved training. No special intervention (learning rate decrease, checkpoint rollback, or loss-scale adjustment) was required for either run, supporting our hypothesis that the encoder's pre-norm RMSNorm architecture is broadly sufficient to stabilize routing on its own. Validation accuracy curves for both Soft-MoE variants are provided in Appendix C.

Interpretation. The two regimes we observe in Expert-Choice, severe instability under consecutive placement, milder transient collapses under interleaved placement, map

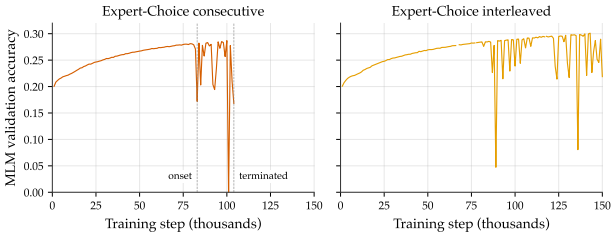

Figure 1. Masked language modeling validation accuracy at masking rate 0.15 for Expert-Choice consecutive (left) and Expert-Choice interleaved (right). Consecutive Expert-Choice trains smoothly until step 83K, then enters large-amplitude oscillations that grow in magnitude; we terminated training at step 104K. Interleaved Expert-Choice exhibits transient downward spikes starting near step 85K from which the model recovers, maintaining a stable upper envelope through step 150K. The same qualitative behavior is observed at masking rates 0.20, 0.25, and 0.30 (appendix).

onto the two corresponding predictions of Section 3.3: cascading discrete flips when discontinuous operations are composed directly, and individual flips followed by a smoothing dense layer when they are not. We interpret the Expert-Choice consecutive divergence and the Expert-Choice interleaved transient collapses as two manifestations of the same underlying property, the discontinuity of top-k selection, differing in how many discontinuous operations are composed without a smoothing intermediate. Section 3.3's third prediction, that Soft-MoE is placement-invariant, is consistent with the stability we observe for the Soft-MoE consecutive variants; we restrict our matched-compute comparison to consecutive placement and use the available budget for the L_2 ablation under that configuration. We refer to Section 3.3 for the structural argument and do not restate it here.

4.2. Downstream Benchmarks

We evaluate on six established benchmarks with standardized splits from TAPE, FLIP, SCOP, and ProteinGym, covering local structure, global structural organization, thermodynamic properties, and functional fitness landscapes. We report secondary structure on both CASP12 (2016) and CASP14 (2020); the substantially larger CASP12-to-CASP14 gap exhibited by the dense baselines (4.6–6.3 points) compared to Ares (0.4 points) is consistent with the possibility that partial CASP12 sequences entered the dense baselines' pretraining sets, making CASP14 the cleaner benchmark for cross-model generalization.

We evaluate all models with a frozen backbone and a lightweight task-specific head: a two-linear-layer MLP on mean-pooled residue embeddings, with the form $\mathrm{Linear}(d_{\text{embed}}, 768) \rightarrow \mathrm{GELU} \rightarrow \mathrm{LayerNorm}(768) \rightarrow \mathrm{Linear}(768, n_{\text{classes}})$. We fix the projection dimension at 768 because it is the smallest embedding dimension among the compared backbones (Ankh Base; $d = 768$); pinning

Table 1. Downstream benchmark performance with frozen backbones and a lightweight MLP probe (mean ± std over 3 seeds). Metrics: fluorescence Spearman ρ (Flu.), GB1 fitness Spearman ρ (GB1, FLIP `two_vs_rest` split), remote homology top-1 fold-level accuracy (Rem. Hom., SCOP), stability Spearman ρ (Stab.), secondary structure 3-state accuracy on CASP12 (SS3-C12) and CASP14 (SS3-C14), and ProteinGym aggregate Spearman $\rho \times 100$ across all five functional categories. Higher is better for all metrics. All values reported as percentages. Best result per column is in **bold**, second-best is underlined. Ares rows report the three trained variants retained for downstream evaluation; the Ares Expert-Choice consecutive checkpoint is excluded due to training divergence (Section 4.1).

Model	Flu. ($\rho \uparrow$)	GB1 ($\rho \uparrow$)	Rem. Hom. ($\uparrow$)	Stab. ($\rho \uparrow$)	SS3-C12 ($\uparrow$)	SS3-C14 ($\uparrow$)	ProteinGym ($\rho \uparrow$)
ESM-2 650M	53.30 ± 0.10	44.90 ± 2.09	31.20 ± 0.37	70.78 ± 3.70	81.42 ± 0.27	76.51 ± 0.68	**41.40**
ESM-2 3B	53.90 ± 0.10	44.86 ± 2.52	$\mathbf{34.08 \pm 0.35}$	76.82 ± 1.98	81.97 ± 0.13	75.63 ± 0.41	40.60
Ankh Base	57.90 ± 0.10	45.80 ± 1.55	30.27 ± 0.08	74.14 ± 2.54	80.55 ± 0.08	75.93 ± 0.08	25.20
Ankh Large	$\mathbf{60.80 \pm 0.00}$	$\mathbf{47.80 \pm 1.41}$	19.82 ± 0.08	$\mathbf{79.21 \pm 0.77}$	$\mathbf{82.92 \pm 0.13}$	$\mathbf{77.22 \pm 0.13}$	37.70
Ares EC (interleaved)	54.10 ± 0.20	46.80 ± 0.83	20.98 ± 0.16	57.36 ± 2.23	73.52 ± 0.31	69.70 ± 0.48	12.40
Ares Soft (no L_2)	55.40 ± 0.10	36.80 ± 8.03	25.95 ± 0.16	73.62 ± 0.39	74.67 ± 0.21	74.32 ± 0.27	32.50
Ares Soft (L_2)	59.00 ± 0.10	45.00 ± 0.75	26.51 ± 0.16	66.60 ± 2.44	74.30 ± 0.19	73.88 ± 0.22	30.30

the probe width to the smallest baseline removes one source of probe-capacity inflation across models and is decidable before any benchmark results are observed. Because the first linear still maps from each model's native hidden dimension to this fixed projection size, higher-dimensional backbones receive a wider input-side projection than Ares ($d = 1024$): ESM-2 3B ($d = 2560$) in particular has approximately $2.5\times$ the first-layer probe parameters of Ares. The comparison therefore evaluates Ares against dense baselines that benefit both from a higher-capacity probe and from substantially more pretraining tokens. Per-category ProteinGym breakdown is in Appendix A. To verify that the 150K-step comparison is not artificially constrained by undertraining, we extended Soft-MoE L_2 by an additional 75K steps; Appendix D shows consistent improvement across all ProteinGym categories, indicating the 150K-step results reflect an undersaturated regime.

Scope of comparison. Our experiments target the routing-strategy and placement comparisons under matched compute; we situate the resulting models against publicly released dense encoders (ESM-2 650M, ESM-2 3B, Ankh Base, Ankh Large) as external reference points. The absolute differences in Table 1 therefore reflect a combined effect of architecture and training budget, while the internal MoE-vs-MoE comparisons remain the controlled contrast.

4.3. Interpretability Analysis

The structural argument of Section 3.3 predicted that the discreteness of Expert-Choice routing and the smoothness of Soft-MoE routing would produce qualitatively different internal behavior. We examine the trained checkpoints directly to test this prediction. Two contrasts organize this section: the discrete-versus-smooth comparison (Section 4.3.1, comparing Expert-Choice to Soft-MoE) and the L2 ablation revisited at the level of expert specialization (Section 4.3.2, comparing the two Soft-MoE variants). We focus on findings most relevant to interpreting the benchmark results in Table 1; additional figures and per-layer breakdowns are deferred to the appendix.

4.3.1. Routing Discreteness and Layer-Level Collapse

We probe the Ares Expert-Choice interleaved checkpoint (which trained to 150K steps despite the transient collapses documented in Section 4.1) using three complementary measurements at each MoE layer: pairwise expert co-selection (Jaccard overlap of selected token sets), drop rate (fraction of tokens not selected by any expert under the capacity-constrained top-k assignment), and single-expert knockout (the change in masked-language-modeling loss when a single expert is replaced with zeros).

Layer 7 exhibits expert collapse. Across the ten MoE layers in the interleaved configuration (L1, L3, L5, …, L19), nine show the routing structure expected of a healthy MoE layer: low pairwise co-selection, modest drop rates, and visible block structure indicating subgroups of experts that activate together. Layer 7, by contrast, is a clear outlier on every measurement (Figures 2 and 3). Its pairwise co-selection matrix is uniformly high, with the majority of expert pairs sharing more than half of their selected tokens, and its drop rate is more than double that of any other layer. The single-expert knockout analysis confirms the diagnosis: removing any single expert at L7 changes the masked-language-modeling loss by less than one percent of baseline, nearly half of the experts have negative knockout deltas (removing them slightly decreases loss), and the mean knockout effect across the layer is statistically indistinguishable from zero. By all three measurements, L7 has degenerated into a regime where the experts compute approximately interchangeable functions and the routing decision contributes little to the layer's output.

Connection to the stability story. The L7 collapse pattern is consistent with a residual transient routing-collapse event of the type documented in Section 4.1: a single layer driven into a degraded routing regime by a discrete flip, while the

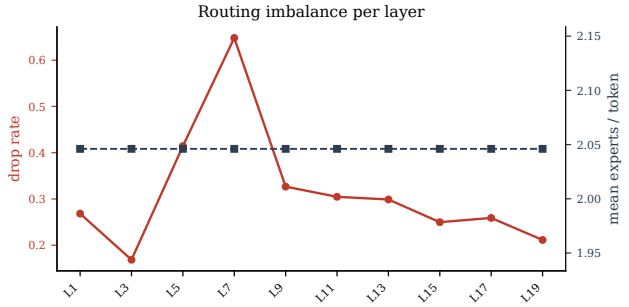

Figure 2. Per-layer token drop rate across the ten MoE layers of the Ares Expert-Choice interleaved checkpoint. Drop rate is the fraction of tokens not selected by any expert under the capacity-constrained top-k assignment. Layer 7 is a clear outlier, with a drop rate more than double that of any other MoE layer.

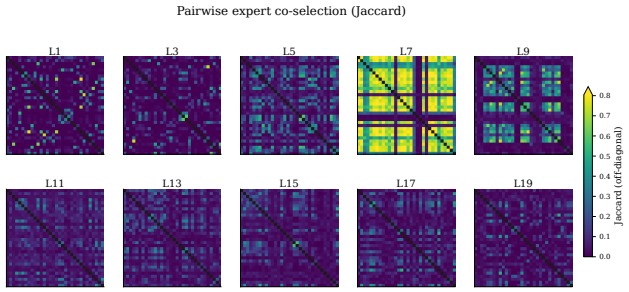

Figure 3. Pairwise expert co-selection (Jaccard overlap of selected token sets) across the ten MoE layers of the Ares Expert-Choice interleaved checkpoint. Most layers exhibit low off-diagonal Jaccard with visible block structure indicating subgroups of experts that activate together. Layer 7 is uniformly bright, with the majority of expert pairs sharing more than half of their selected tokens.

rest of the network continues to train normally. The interleaved configuration prevents the cascading divergence observed in the consecutive case but does not eliminate the underlying discontinuity of top-k routing, an individual flip at a single layer, if severe enough, can leave that layer in a state from which the model does not recover; a possible architectural mitigation for the final layer is noted in Appendix B. The model compensates by routing meaningful work through other layers, which is why downstream representations remain usable, but the price is visible in Table 1 and in Appendix A, where the Expert-Choice interleaved variant exhibits the lowest scores in our evaluation, with a particularly low Stability subscore on ProteinGym.

4.3.2. The L_2 Ablation Revisited: Dispatch Collapse and Combine-Only Routing

Section 3.2 motivated the router-side L_2 normalization ablation with two hypotheses: (i) the encoder's pre-norm RMSNorm makes router-side L_2 redundant for stability, and (ii) unnormalized tokens and slots preserve magnitude information that may support stronger expert specialization. Sections 4.1 and 4.2 support (i): both Soft-MoE variants

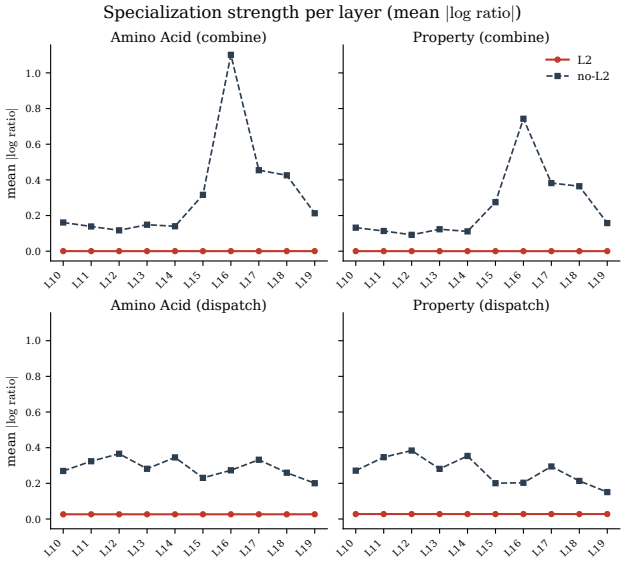

Figure 4. Per-layer mean $|\log \text{ratio}|$ across the ten MoE layers, comparing the L_2 and no-L_2 Soft-MoE variants on amino-acid and property routing for both dispatch and combine. The no-L_2 variant produces values one to several orders of magnitude larger across all four facets. Under L_2 normalization, routing logits are bounded scaled cosine similarities, which mechanically caps the achievable $|\log \text{ratio}|$ by construction.

train stably and achieve comparable downstream performance. We now examine (ii) using three views, raw specialization strength, scale-invariant per-expert selectivity, and across-expert specialization, and find a more specific picture than (ii) anticipated: L_2 does not merely weaken dispatch-side specialization, it functionally disables expert-level distinction at the dispatch step while leaving combine intact, reducing the MoE layer to a parallel ensemble of expert MLPs over a near-shared input pool with token-conditioned readout.

Raw specialization strength is much larger without L_2, but this is partially mechanical. For each MoE layer and each expert, we compute the log-ratio of how often that expert routes a particular amino acid (or biochemical property) relative to a uniform-routing baseline. Across all four facets, amino acid and property; dispatch and combine, the no-L_2 variant produces $|\log \text{ratio}|$ values one to several orders of magnitude larger than the L_2 variant (Figure 4). This raw comparison, however, is partially structural rather than learned: under L_2 normalization, routing logits are scaled cosine similarities bounded in $[-s, +s]$ for a learnable scalar s, which mechanically caps the maximum achievable $|\log \text{ratio}|$ by construction. The unnormalized variant has no such bound. The strength gap in Figure 4 therefore reflects what the parameterization can express, not how strongly each variant chooses to specialize.

On the scale-invariant comparison, L_2 is at least as se-

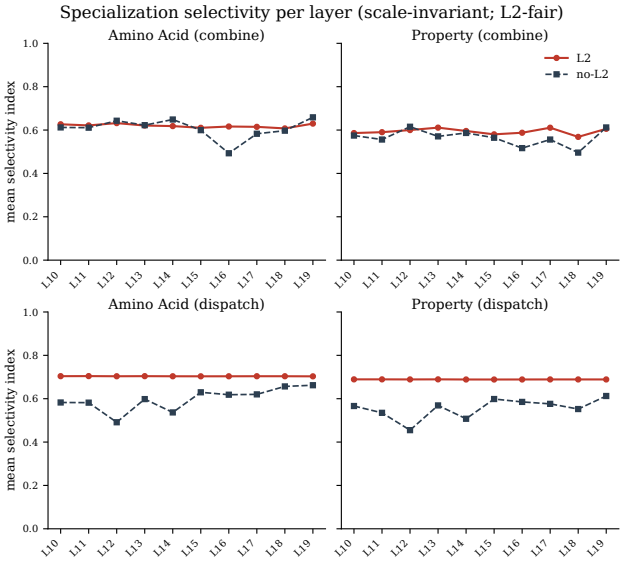

Figure 5. Per-layer scale-invariant selectivity ($1 - \text{mean}/\text{max}$ of $|\log \text{ratio}|$ per expert, averaged within each layer; values in $[0, 1]$). L_2 and no-L_2 are comparable on combine and L_2 is noticeably higher on dispatch, the opposite of what the raw strength comparison in Figure 4 would suggest. Per-expert selectivity does not imply expert-level differentiation across the layer.

lective per expert as no-L_2. To compare specialization in a way that does not reflect parameterization bounds, we report a scale-invariant selectivity metric: $1 - \text{mean}/\text{max}$ of $|\log \text{ratio}|$ per expert, averaged within each layer. This metric lies in $[0, 1]$ and measures how concentrated each expert's preference distribution is *relative to its own range*, independent of absolute magnitude. The four panels in Figure 5 show that L_2 selectivity is comparable to no-L_2 on combine and noticeably higher than no-L_2 on dispatch. Under the parameterization-fair comparison, individual L_2 experts are at least as concentrated in their preferences as no-L_2 experts, and on dispatch they are more concentrated. As we show next, this individual-expert concentration coexists with a complete collapse of expert-level distinction across the layer.

Dispatch-side expert distinction collapses under L_2; combine-side expert structure is preserved. A 32-expert layer in which every expert is individually selective but all experts share the same preference is selective per expert and yet has no expert-level differentiation as a layer. We examine two across-expert measurements: *expert diversity*, the number of distinct argmax classes among the 32 experts in each MoE layer, and *expert redundancy*, the mean within-layer cosine similarity of expert preference vectors. On dispatch, the L_2 variant exhibits a degenerate regime: every layer collapses to a single distinct argmax class, with within-layer expert cosine similarity saturating at 1.0 across all ten MoE layers and across both class systems. All 32 experts in every layer share the same preference profile, and so receive a near-identical mixture of incoming tokens. The no-L_2 dispatch experts, in contrast, cover multiple distinct classes per layer with sub-saturating cosine similarity. The dispatch step in the L_2 variant is therefore not a softer version of dispatch in the no-L_2 variant; expert-level distinction has been mechanically removed. On combine, the picture is qualitatively different. The L_2 checkpoint shows decorrelated experts and broad coverage of the available class space, comparable to the no-L_2 checkpoint on the same axis. Dispatch and combine therefore do not collapse together: dispatch is collapsed, combine is not. Per-layer numerical values for both measurements are reported in Appendix E.

Why dispatch and combine decouple. Both are softmaxes over the same logit tensor produced by the router, dispatch normalizing along the token axis and combine normalizing along the joint expert–slot axis, so the bound that L_2 normalization places on logit magnitudes applies symmetrically to the two and does not predict the asymmetry. What does predict it is that under L_2 training the slot anchors converge toward a near-shared direction (the saturating dispatch-side cosine similarity is the direct empirical signature of this). When slot anchors cluster, every expert's column of routing logits has the same relative ranking over tokens, so all experts receive the same dispatch profile and dispatch-side expert distinction collapses. The combine-side row-softmax operates within a single token's row and depends only on the relative ordering of slot logits within that row; small per-slot variation in slot anchors, combined with distinct per-expert MLPs, is enough to produce a different per-token mixture for each token, leaving the combine side differentiated.

The L_2 MoE layer reduces to a combine-only computation. Putting these observations together, the L_2 MoE layer is not performing token-level routing in the conventional sense. With dispatch-side expert differentiation fully collapsed, every expert receives a near-identical mixture of incoming tokens; the differentiation across experts comes entirely from their distinct MLPs applied in parallel to that shared input pool, and the per-token output is a token-specific weighted readout produced by the intact combine step. L_2 does not merely constrain or reshape dispatch; it removes dispatch from the routing role entirely, leaving the combine step as the sole source of per-token differentiation in the layer. Cross-run agreement between L_2 and no-L_2 is correspondingly low, especially on combine, suggesting that combine-side per-expert class assignments are not strongly determined by the data alone in either variant (Appendix E).

This mechanical difference does not translate into a benchmark advantage. Across the six downstream tasks in Table 1, L_2 wins decisively on Fluorescence and GB1, no-L_2 wins decisively on Stability and ProteinGym, and the

two are essentially tied on Remote Homology and on both secondary-structure benchmarks. The order-of-magnitude differences in dispatch-side specialization do not produce a corresponding differentiation in benchmark performance at this scale and token budget. The implication is that the dispatch step in Soft-MoE may contribute less to representation quality, in this regime, than the standard Soft-MoE story predicts: a model whose dispatch step has been mechanically disabled remains competitive with one whose dispatch step is highly differentiated.

Implications for router design. If a Soft-MoE configuration can disable its dispatch step and still retain downstream performance, the natural question is whether the dispatch step is recovering anything that justifies its parameter complexity. Two directions follow. The first is to train without router-side L_2 but with intermediate bounds on the dispatch logits, soft capping (e.g., $\tanh$-based bounds applied to unnormalized inner products, as used in recent large language models), unilateral normalization (normalizing only tokens or only slots), or learnable softmax temperature, applied differentially to dispatch versus combine since the constraint binds asymmetrically. The second is to test whether a deliberately combine-only Soft-MoE, with dispatch replaced by uniform averaging by construction, matches or improves on the L_2 variant. We note that our combine-side measurements come from a single trained checkpoint per variant, so we cannot rule out that the combine-side differentiation we observe partly reflects amplification of small residual slot-anchor variation rather than learned class-level specialization; multi-seed verification is left to future work alongside the design alternatives above.

5. Conclusion

We presented Ares, the first application of loss-free Mixture-of-Experts routing to bidirectional protein encoders, and a controlled comparison of Soft-MoE and Expert-Choice routing under identical data and compute. Three findings emerged. First, the two routing strategies are not stability-equivalent: consecutive Expert-Choice diverges, interleaved Expert-Choice exhibits transient routing collapses, and Soft-MoE trains stably regardless of placement. We argued that this follows from applying a known property of top-k routing, the gradient mismatch introduced by piecewise-constant selection under a straight-through estimator, to the depth dimension: when discrete routing operations are composed without a smoothing intermediate, individual routing flips can compound across layers rather than being absorbed by surrounding smooth transformations. Second, in the trained Expert-Choice interleaved checkpoint we identified a single-layer expert collapse (L7) consistent with a residual transient routing-collapse event, indicating that interleaving prevents cascading divergence without eliminating the underlying discontinuity. Third, ablating router-side L_2 normalization in Soft-MoE reveals that L_2 functionally disables the dispatch step, every dispatch layer collapses to a single argmax class with saturated within-layer expert similarity, while combine retains substantial expert structure, reducing the MoE layer to a parallel ensemble of expert MLPs over a near-shared input pool with token-conditioned readout; this combine-only configuration matches the no-L_2 variant's downstream performance, suggesting that the dispatch step in Soft-MoE contributes less to representation quality at this scale than the standard Soft-MoE story predicts. Within our 39B-token budget, continuous routing is a more robust foundation for MoE-based protein encoders than discrete top-k routing, and constraints on router geometry can disable individual routing components without a corresponding penalty on downstream tasks. Larger token budgets are a natural next step for characterizing how the routing-strategy comparison and the dispatch-versus-combine contribution behave at scale.

Impact Statement

This paper studies routing strategies for Mixture-of-Experts protein language models. Like other work on protein language modeling, the resulting representations support downstream applications including variant-effect prediction, structure prediction, and protein design, and we acknowledge the well-documented dual-use considerations that accompany this broader research area. Our methodological contribution is one of compute efficiency: Mixture-of-Experts decouples model capacity from per-token compute, with a favorable environmental profile relative to scaling dense models to comparable parameter counts. We release training code, evaluation pipelines, and model weights to support reproducibility and to enable the broader community to audit and build on this work.

Acknowledgments

We gratefully acknowledge Google's TPU Research Cloud (TRC) program for providing the Cloud TPU resources that made the training of Ares possible. We thank Dr. Ahmed Saleh Mansour for his valuable feedback and careful review of the manuscript.

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Ankner, Z., Saphra, N., Blalock, D., Frankle, J., and Leavitt,

M. L. Dynamic masking rate schedules for mlm pretraining, 2024. URL https://arxiv.org/abs/2305.15096.

Antoniak, S., Krutul, M., Pióro, M., Krajewski, J., Ludziejewski, J., Ciebiera, K., Król, K., Odrzygóźdź, T., Cygan, M., and Jaszczur, S. Mixture of tokens: Continuous moe through cross-example aggregation, 2024. URL https://arxiv.org/abs/2310.15961.

Bhatnagar, A., Jain, S., Beazer, J., Curran, S. C., Hoffnagle, A. M., Ching, K. S., Martyn, M., Nayfach, S., Ruffolo, J. A., and Madani, A. Scaling unlocks broader generation and deeper functional understanding of proteins. *bioRxiv*, pp. 2025–04, 2025.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Elnaggar, A., Essam, H., Salah-Eldin, W., Moustafa, W., Elkerdawy, M., Rochereau, C., and Rost, B. Ankh: Optimized protein language model unlocks general-purpose modelling, 2023. URL https://arxiv.org/abs/2301.06568.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding, 2020. URL https://arxiv.org/abs/2006.16668.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. Progen2: Exploring the boundaries of protein language models, 2022. URL https://arxiv.org/abs/2206.13517.

Puigcerver, J., Riquelme, C., Mustafa, B., and Houlsby, N. From sparse to soft mixtures of experts, 2024. URL https://arxiv.org/abs/2308.00951.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts, 2021. URL https://arxiv.org/abs/2106.05974.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 05 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm098. URL https://doi.org/10.1093/bioinformatics/btm098.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., and Poli, I. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.

Wettig, A., Gao, T., Zhong, Z., and Chen, D. Should you mask 15 URL https://arxiv.org/abs/2202.08005.

Zhang, B. and Sennrich, R. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A., Chen, Z., Le, Q., and Laudon, J. Mixture-of-experts with expert choice routing, 2022. URL https://arxiv.org/abs/2202.09368.

A. ProteinGym Per-Category Breakdown

Table 1 reports the ProteinGym aggregate Spearman correlation across all five functional categories. Table 2 provides the per-category breakdown across Activity, Binding, Expression, Organismal Fitness, and Stability subsets, evaluated under the same frozen-backbone MLP-probe protocol used in the main results.

The breakdown reveals heterogeneous behavior that the aggregate hides. Both Soft-MoE variants beat Ankh Base on four of five categories (Activity, Binding, Expression, Organismal Fitness), with Ankh Base maintaining its lead only on Stability. On Binding specifically, both Soft-MoE variants (30.50 and 28.10) exceed Ankh Large (26.50). The Expert-Choice interleaved variant performs uniformly poorly across all five categories, with a particularly low Stability subscore (6.50), which we interpret as further evidence that the residual routing instability documented in Section 4.1 degrades representation quality on stability-related variant-effect prediction. Across all categories, ESM-2 650M and ESM-2 3B retain a consistent lead, reflecting the substantially larger pretraining token budget of these dense baselines.

Table 2. ProteinGym per-category breakdown. Spearman $\rho \times 100$ on each of the five functional categories defined by ProteinGym, plus the aggregate (All) reproduced from Table 1. Higher is better. Best result per column is in **bold**, second-best is <u>underlined</u>.

Model	Activity ($\rho \uparrow$)	Binding ($\rho \uparrow$)	Expression ($\rho \uparrow$)	Org. Fitness ($\rho \uparrow$)	Stability ($\rho \uparrow$)	All ($\rho \uparrow$)
ESM-2 650M	**42.50**	**33.70**	**41.50**	<u>36.80</u>	**52.30**	**41.40**
ESM-2 3B	<u>41.70</u>	<u>32.10</u>	<u>40.30</u>	**37.80**	<u>50.90</u>	<u>40.60</u>
Ankh Base	20.80	13.40	24.80	17.70	49.40	25.20
Ankh Large	38.60	26.50	38.90	35.10	49.20	37.70
Ares EC (interleaved)	13.90	13.40	17.60	10.70	6.50	12.40
Ares Soft (no L_2)	33.00	30.50	36.40	24.60	37.90	32.50
Ares Soft (L_2)	30.40	28.10	33.10	22.60	37.50	30.30

B. Absorbing the final routing output before the LM head.

Our interleaved configuration places dense layers between Expert-Choice layers, but leaves the final network layer as an Expert-Choice layer feeding directly into the LM head. One plausible mitigation we did not evaluate, due to compute constraints, is placing an additional dense feed-forward layer immediately before the LM head. The mechanism here is distinct from the one our interleaved configuration already addresses: rather than interrupting the composition of discontinuous operations across depth, the additional dense FFN would give the network learned non-routed capacity to re-mix the output of the final Expert-Choice layer before the LM head. Being continuous, this layer cannot eliminate the discontinuity from a last-layer routing flip, but it offers capacity to learn a representation that is less sensitive to the final expert assignment, dampening the magnitude of the resulting loss perturbation rather than passing the raw routing output directly to the head.

C. Soft-MoE Pretraining Stability

Section 4.1 reports that both Soft-MoE consecutive variants, with router L_2 normalization and without, train through the full schedule without the divergent oscillations of Expert-Choice consecutive or the recurrent collapses of Expert-Choice interleaved. Figure 6 presents validation accuracy at curriculum masking rate 0.15 for both variants over the 150K-step training run.

The two variants follow comparable trajectories through the first $\sim$80K steps, after which the no-L_2 variant tracks slightly above the L_2 variant for the remainder of training. The no-L_2 variant exhibits a single transient drop near step 140K, with validation accuracy descending from approximately 0.31 to 0.24 over a few thousand steps (validation perplexity rising from 9.70 at step 139K to 12.38 at step 140K) before recovering to its prior trajectory. The L_2 variant exhibits no comparable event and trains monotonically through the full schedule. We did not investigate the cause of the no-L_2 event; we note only that it is a single isolated event from which the model recovers cleanly, qualitatively distinct from the repeated collapse-and-recover events documented for Expert-Choice interleaved (Figure 1, right). Final validation accuracy at step 150K is 0.313 for no-L_2 and 0.298 for L_2, a gap of $\sim$1.5 percentage points; this small difference in pretraining accuracy does not translate into a consistent downstream advantage for either variant in Table 1.

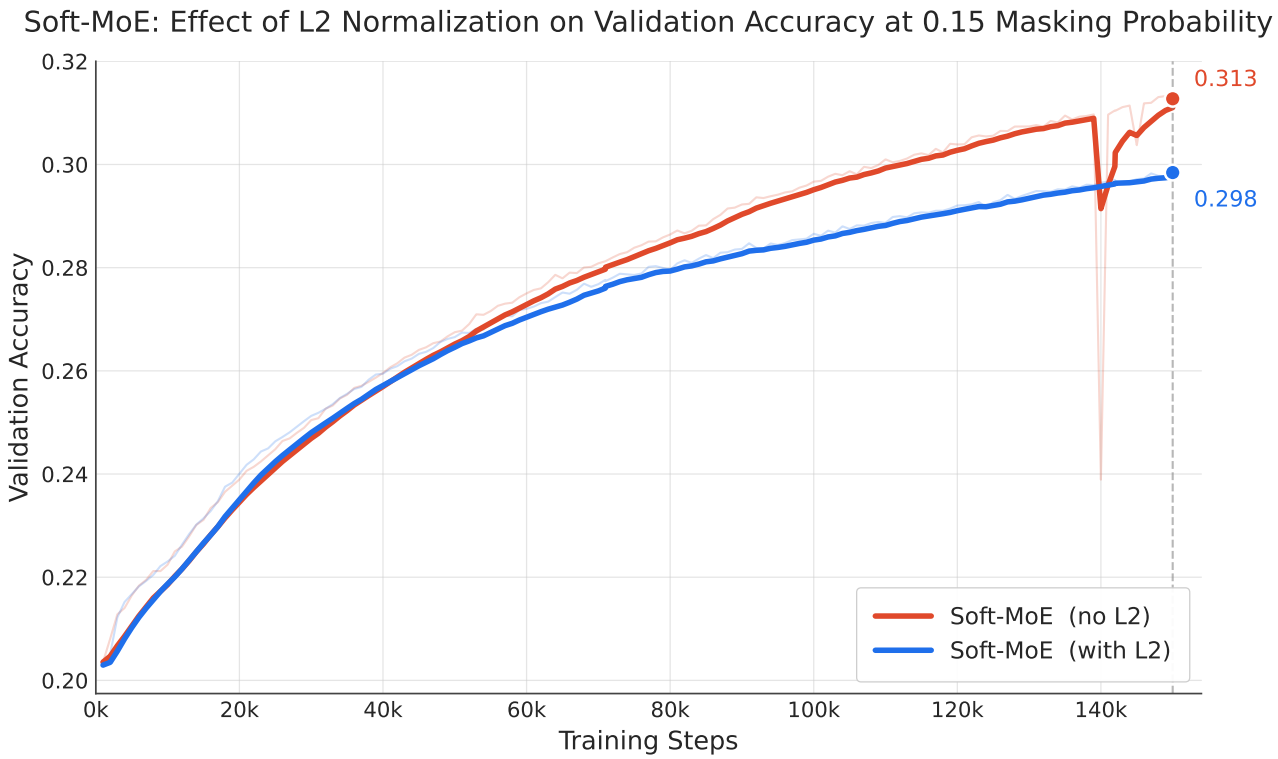

Figure 6. Validation accuracy at curriculum masking rate 0.15 for the two Soft-MoE consecutive variants over 150K training steps. The no-L_2 variant (red) trains slightly above the L_2 variant (blue) from ∼80K onward, exhibiting a single transient drop near step 140K from which the model recovers. The L_2 variant (blue) trains monotonically. Final validation accuracies at step 150K: no-L_2 = 0.313, L_2 = 0.298.

D. Training Saturation Check

The downstream comparison in Table 1 is conducted at a controlled 150K-step budget across all three Ares variants, ensuring a like-for-like routing comparison. A separate question is whether 150K steps reflects a converged training regime or an undersaturated one, if the latter, the absolute performance levels in Table 1 understate what the same architecture would reach with additional compute. To probe this, we extended one of the three variants (Soft-MoE consecutive with router L_2 normalization) by an additional 75K steps, to a total of 225K, and re-evaluated it on ProteinGym under the same frozen-backbone MLP-probe protocol used in the main results.

The 225K checkpoint improves on the 150K checkpoint in every ProteinGym category, with no regression on any subset (Table 3). The aggregate improves by 2.0 points (30.30 to 32.33), and Stability, the category with the largest 150K-step difference between Soft-MoE L_2 and Ankh Base, improves by 3.7 points. The validation-accuracy trajectory of the full 225K run (Figure 7) shows that pretraining accuracy continues rising through 225K without plateauing, consistent with the downstream improvements observed on ProteinGym. The 150K-step controlled comparison therefore reflects an undersaturated training regime, and the absolute downstream numbers in Table 1 should be read accordingly.

This saturation check is conducted on one variant and characterizes the 150K comparison as undersaturated; it indicates the direction in which absolute performance moves with additional compute, while the controlled MoE-vs-MoE comparison at 150K remains the primary contrast.

Table 3. Training saturation check on Soft-MoE consecutive (L_2). ProteinGym Spearman $\rho \times 100$ at 150K (the controlled-comparison checkpoint used in Table 1) and at 225K (after an additional 75K training steps). Higher is better; positive Δ indicates improvement.

Category	150K	225K	Δ
Activity	30.40	32.37	+1.97
Binding	28.10	28.57	+0.47
Expression	33.10	35.44	+2.34
Organismal Fitness	22.60	24.03	+1.43
Stability	37.50	41.22	+3.72
All	**30.30**	**32.33**	**+2.03**

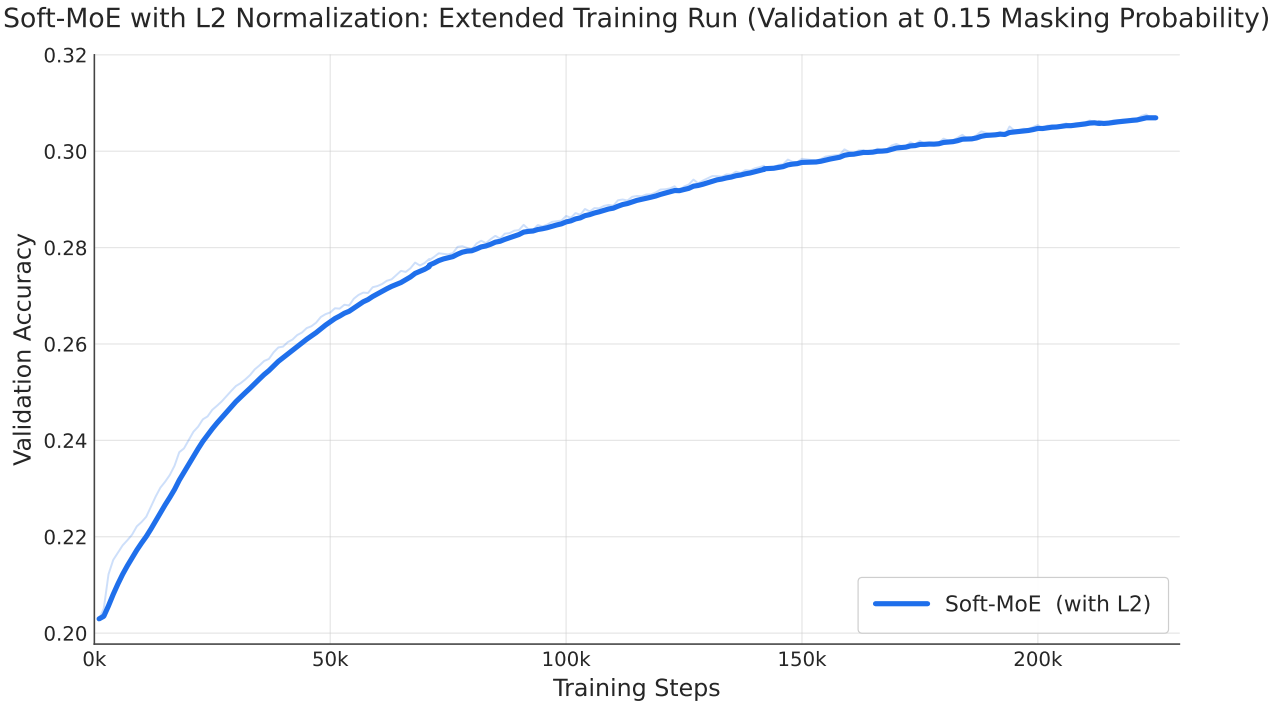

Figure 7. Validation accuracy at curriculum masking rate 0.15 for the Soft-MoE consecutive (L_2) variant over the full 225K-step extended training run. The curve continues rising through 225K without plateauing, supporting the interpretation that the 150K-step controlled comparison in Table 1 reflects an undersaturated training regime.

E. Per-Layer Specialization Measurements

Section 4.3.2 reports per-layer measurements for two across-expert quantities (*expert diversity* and *expert redundancy*) and a cross-run quantity (Spearman correlation between L_2 and no-L_2 routing patterns). This appendix provides the corresponding per-layer plots for completeness.

Expert diversity. For each MoE layer, we count the number of distinct argmax classes among the 32 experts: each expert is assigned the class for which its $|\log \text{ratio}|$ is largest, and we count how many distinct classes appear. The maximum is bounded by $\min(32, \#\text{classes})$, which is 20 for amino acids and 9 for property groups. Figures 8 and 9 show the result on amino-acid and property routing respectively. The dispatch panels visualize the constraint-induced collapse described in Section 4.3.2: L_2 dispatch layers all collapse to a single argmax class, while no-L_2 dispatch experts cover roughly 10 distinct amino-acid classes and 5 property classes per layer on average. The combine panels show that the two variants are broadly comparable, with L_2 covering 14–17 amino-acid classes per layer and no-L_2 covering 10–17 (with no-L_2 dipping to 10–12 around L16–L17), and both covering 5–8 property classes per layer.

Expert redundancy. For each MoE layer, we measure the mean off-diagonal cosine similarity among the 32 expert

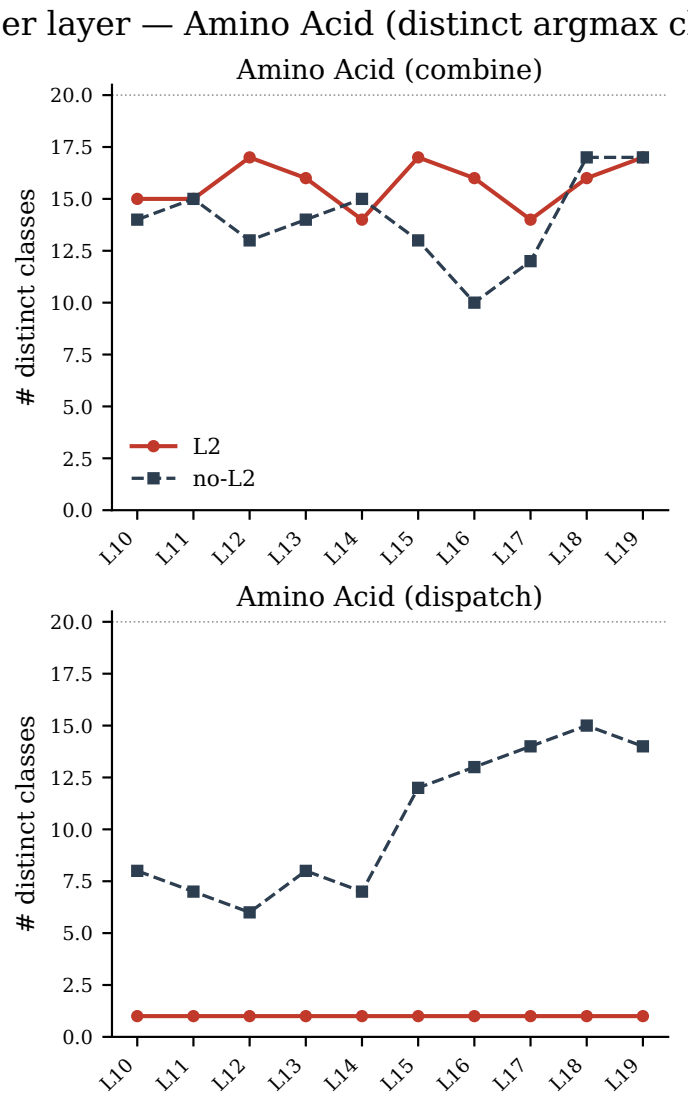

Figure 8. Per-layer expert diversity for amino-acid routing: number of distinct argmax classes among the 32 experts in each MoE layer (max = 20). Top: combine. Bottom: dispatch. The dispatch panel shows the L_2 collapse to a single argmax class across all ten MoE layers; no-L_2 recovers roughly 10 distinct classes per layer on average.

preference vectors (where each expert's preference vector is its $|\log \text{ratio}|$ profile across classes). Higher values indicate more redundant routing; lower values indicate more diverse expert behavior. Figure 10 shows that on dispatch, L_2 saturates at 1.0 across all ten MoE layers (every expert's preference profile is identical, consistent with the collapse to a single argmax class), whereas no-L_2 peaks in early MoE layers (L11–L14, $\sim$0.85–0.95) and falls to $\sim$0.1 in late layers. On combine, L_2 sits near zero throughout; no-L_2 is also near zero in most layers, with a localized spike at L16 ($\sim$0.38 on amino-acid combine, $\sim$0.15 on property combine) that does not appear in L_2.

Cross-run agreement. Figure 11 reports per-layer Spearman correlations between the flattened L_2 and no-L_2 (expert $\times$ class) preference matrices. The agreement is low across all four facets: per-layer Spearman correlations average between 0.05 and 0.32, and 40–95% of experts choose a different most-preferred class between the two runs. On dispatch this is a direct expression of the L_2 collapse documented in Section 4.3.2: the L_2 dispatch profiles share a single argmax class while the no-L_2 dispatch profiles cover many. On combine, the two variants exhibit comparable expert diversity but their per-expert class assignments do not agree (mean Spearman $\rho = 0.035$ for amino acids and 0.054 for properties), which we read as evidence that combine-side per-expert class assignments are not strongly determined by the data alone in either variant rather than as evidence that one variant has the "correct" specialization.

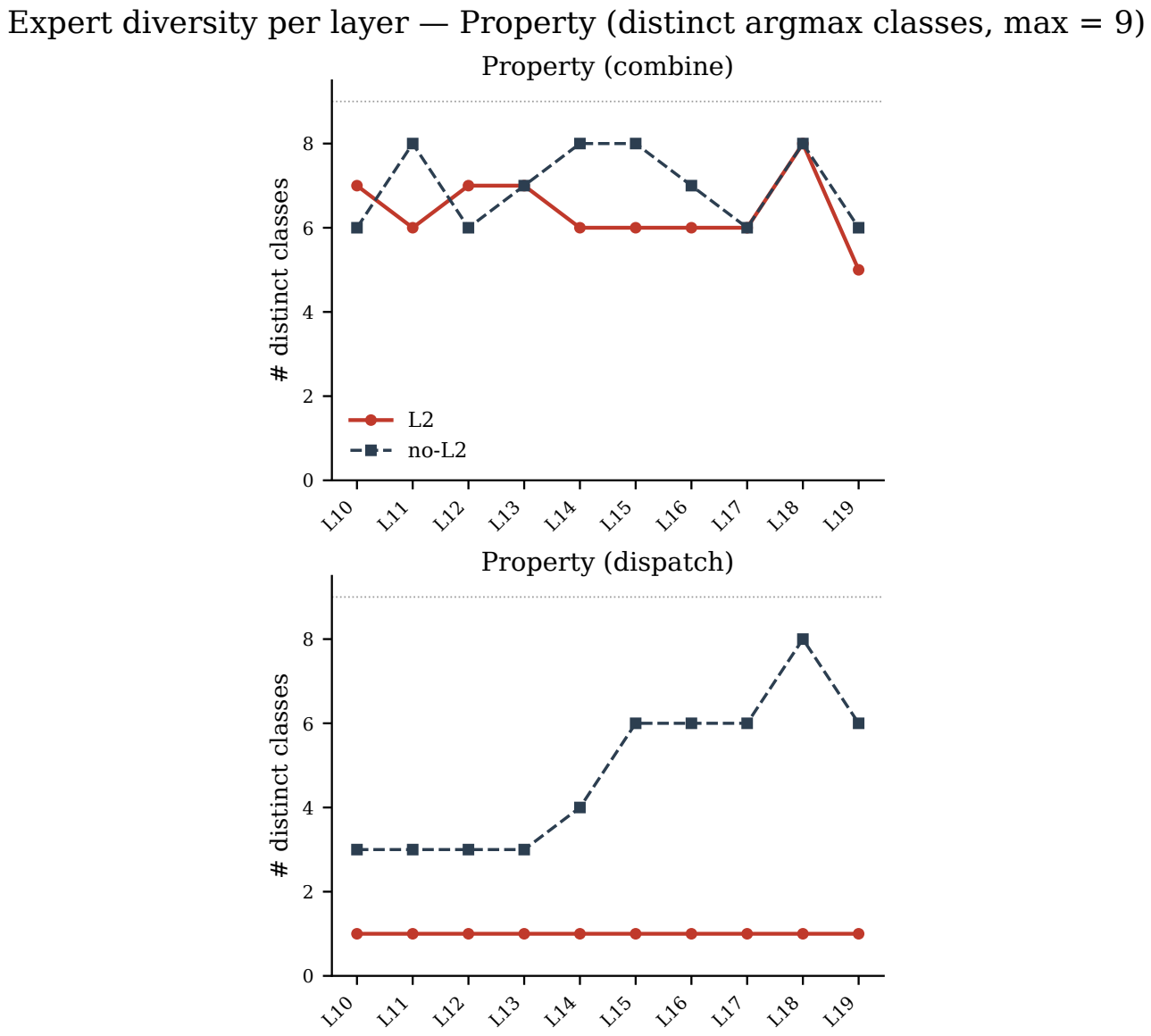

Figure 9. Per-layer expert diversity for property-group routing: number of distinct argmax classes among the 32 experts in each MoE layer (max = 9). Top: combine. Bottom: dispatch. As with the amino-acid view, the L_2 dispatch panel collapses to a single class across all layers, while no-L_2 covers roughly 5 distinct classes per layer on average.

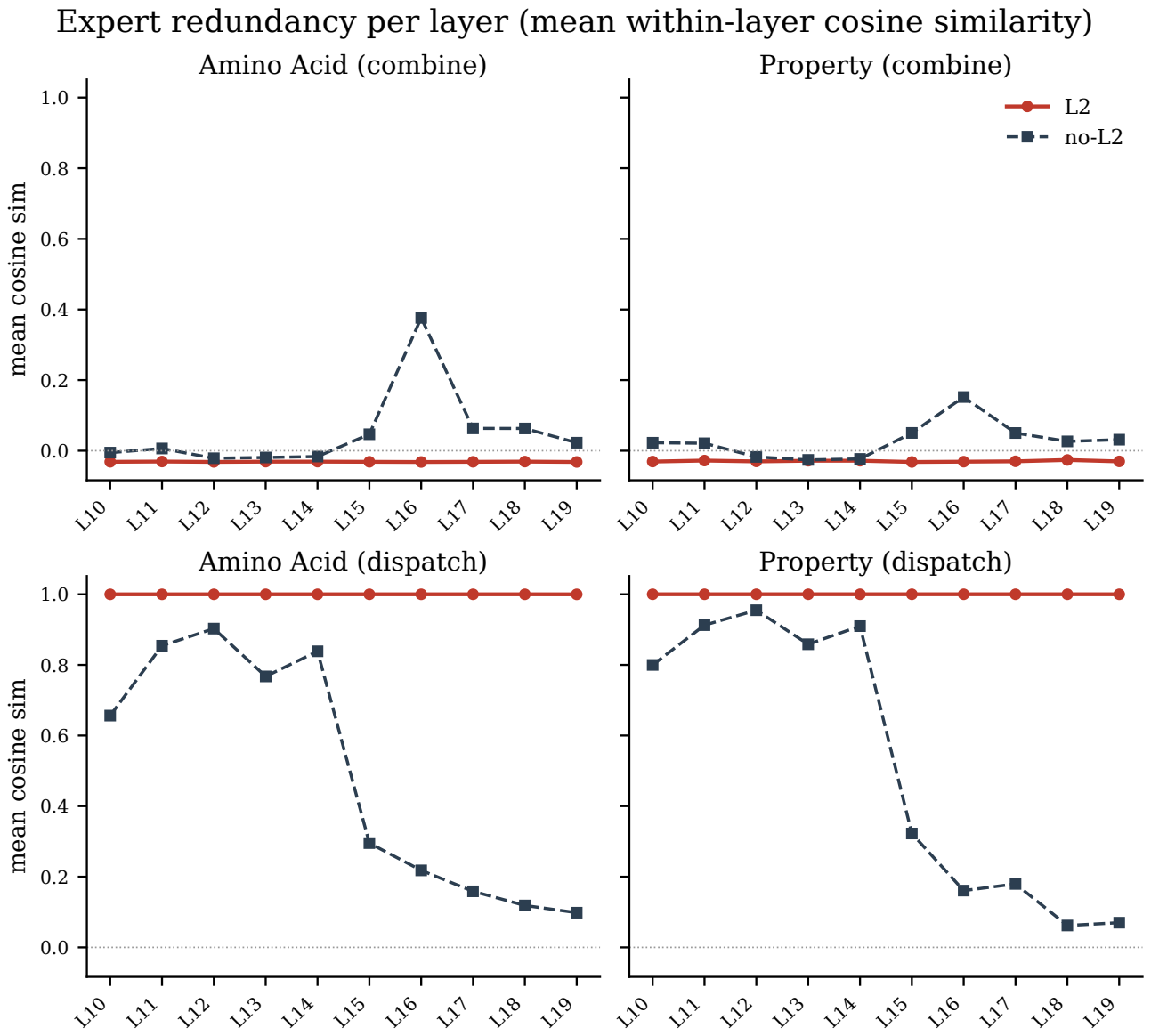

Figure 10. Per-layer expert redundancy: mean within-layer cosine similarity of expert preference vectors. Top row: combine. Bottom row: dispatch. Left: amino-acid routing. Right: property-group routing. The L_2 dispatch line saturating at 1.0 reflects all 32 experts sharing an identical preference profile in each layer.

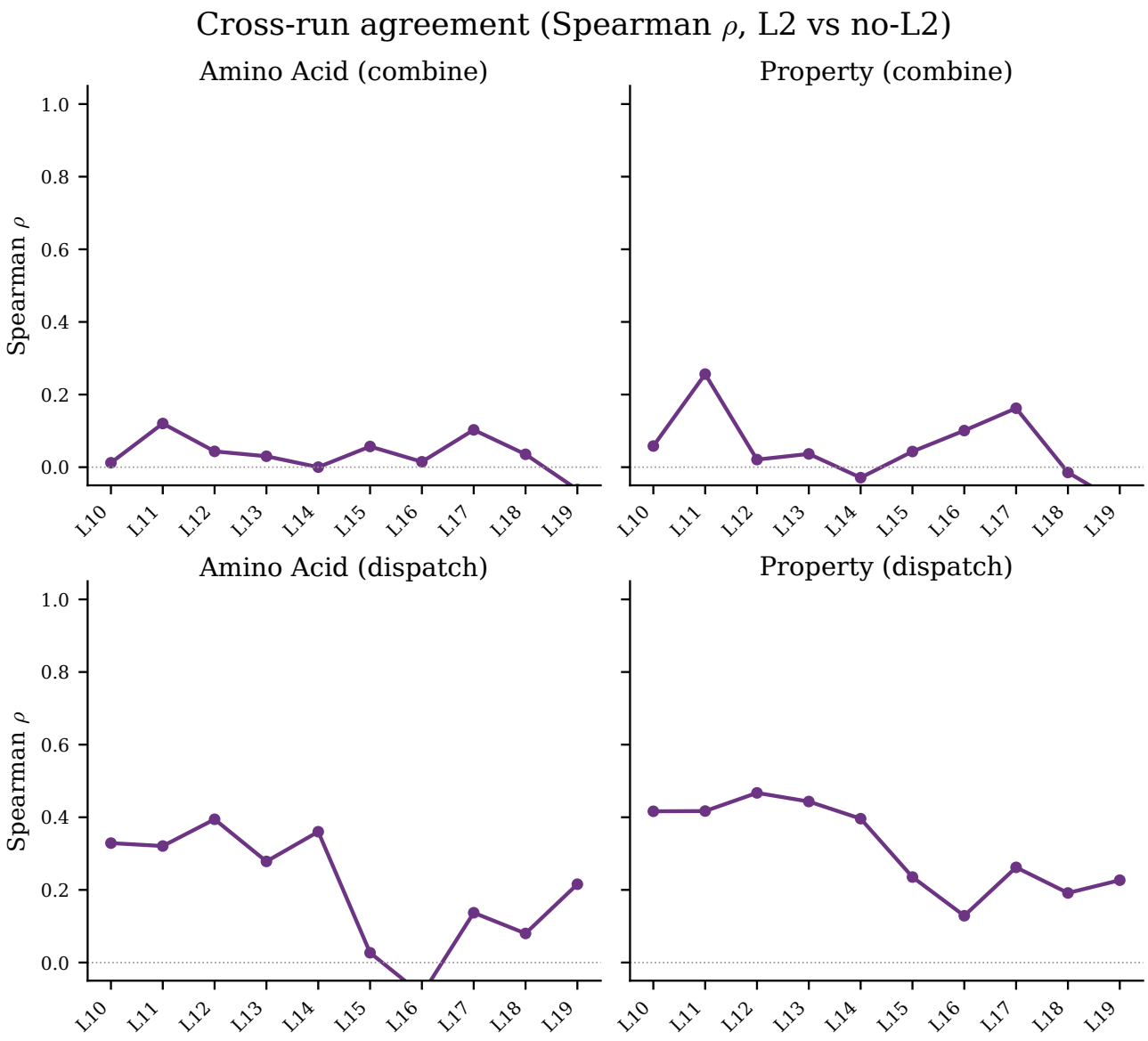

Figure 11. Per-layer Spearman correlation between L_2 and no-L_2 routing patterns, computed on the flattened (expert $\times$ class) $|\log \text{ratio}|$ matrix at each layer. Top row: combine. Bottom row: dispatch. Left: amino-acid routing. Right: property-group routing. Across all four facets the per-layer correlations remain well below 1.0; on dispatch this reflects the L_2 collapse to a single argmax class, on combine it indicates that per-expert class assignments are not strongly data-determined in either variant.